

Masters Program in **Geospatial Technologies**



PALM TREE IMAGE CLASSIFICATION

***A convolutional and machine learning
approach***

Roberto Jose Luna Olivares

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

PALM TREE IMAGE CLASSIFICATION

A convolutional neural network and machine learning approach

Dissertation supervised by

Joel Dinis Baptista Ferreira da Silva, PhD

Instituto Superior de Estatística e Gestão de Informação,

Universidade Nova de Lisboa

Lisbon, Portugal

Dissertation co-supervised by

Pedro da Costa Brito Cabral, PhD

Instituto Superior de Estatística e Gestão de Informação,

Universidade Nova de Lisboa

Lisbon, Portugal

Dissertation co-supervised by

Ignacio Guerrero, PhD

Institute of New Imaging Technologies,

Universitat Jaume I

Castellón de la Plana, Spain

February 2019

ACKNOWLEDGEMENTS

I would like to thanks my supervisor Joel Silva, for sharing his knowledge, encouragement and guidance, made this research project enjoyable and fulfilling. To my co - supervisors Pedro Cabral and Ignacio Guerrero for their feedback and constructive criticism.

Thanks for the effort to all the professor that helped through the master programs. Special thanks to professors Marco Painho and Christoph Brox for their guidance and advice for the duration of the program.To all my classmate that became my family during this period. To the Erasmus Mundus program for giving me the opportunity to complete this master programs and have an unforgettable experience.

To my family for always supporting me and showing me to believe in myself.

*I don't believe it. Prove it to me and
I still won't believe it*

Douglas Adams

PALM TREE IMAGE CLASSIFICATION

A convolutional and machine learning approach

Abstract

Convolutional neural networks have proven to excel at image classification tasks, do to this they have being incorporated into the remote sensing field, initial hurdles in their application like the need for large data sets or heavy computational burden, have being solve with several approaches. In this paper the transfer learning approach is tested for classification of a very high resolution images of a palm oil plantation. This approach uses a pre trained convolutional neural network to extract features from an image, and label them with the aid of machine learning models. The results presented in this study show that the features extracted are a viable option for image classification with the aid of machine learning models. An overall accuracy of 97% in image classification was obtained with the support vector machine model.

KEYWORDS

Convolutional Neural Network

Machine Learning

Unnamed Aerial Vehicle

Image Classification

Transfer Learning

OverFeat

ACRONYMS

GIS - Geographical Information Systems

GEOBIA - Geographic Object-Based Image Analysis

CNN - Convolutional Neural Network

UAV - Unmanned Aerial Vehicle

FVP - First Person View

RC - Remote Control

CMOS - Complementary Metal-Oxide Semiconductor

CHDK - Canon Hack Development Kit

RF - Random Forest

LR - Logistic Regression

DT - Decision Tree

GBC - Gradient Boost Classifier

KNN - K nearest neighbor

SVM - Support Vector Machine

TP - True Positives

TN - True Negatives

FP - False Positives

FN - False Negatives

INDEX OF THE TEXT

INDEX OF FIGURES	x
------------------	---

INDEX OF TABLES	xi
-----------------	----

1 Introduction	1
1.1 An overview of the work	1
1.2 Aim and Objectives	3
1.3 Document organization	3
2 Literature review	4
2.1 Remote sensing and UAV applications	4
2.1.1 Remote sensing	4
2.1.2 UAV applications	5
2.2 Machine learning techniques in Remote Sensing	6
2.3 Convolutional neural-networks	7
2.4 Application of CNN in Remote Sensing	9
2.5 Fundamentals of CNN	10
2.6 OverFeat	11
3 Data and Methods	13
3.1 Description of study area	13
3.2 UAV System	13
3.3 Aerial survey	15
3.4 Data set	15
3.5 Methodology	17
3.5.1 Tools	17
3.6 Data Preprocessing	18
3.7 OverFeat and Feature Extraction	19
3.8 Machine Learning Models	19
3.9 Visualization	20
3.10 Accuracy assessment	20
4 Results	21

4.1	Result overview	21
4.2	Image preprocessing	21
4.3	Feature extraction with OverFeat	22
4.4	Machine learning models classification results	23
4.4.1	SVM classification results	25
4.5	Image Classification	26
5	Discussion and Conclusion	31
5.1	Discussion	31
5.2	Conclusion	34
	Bibliographic References	35
I	Annex	42
I.1	Example of features extracted using OverFeat	42
I.2	Image classification results	42
I.3	ROC graphs	43
I.4	Confusion matrix	45
II	Annex 2	47
II.1	Cropping Images	48
II.2	Extracting features with Overfeat	48
II.3	Machine learning Models	50
II.4	Fine tuning Machine learning Models	51
II.4.1	Support Vector Machine	51

INDEX OF FIGURES

2.1	Artificial neural network architecture	8
2.2	Architecture of a Convolutional Neural Network	11
3.1	Study area in Loma de Mico, East Nicaragua	14
3.2	Fixed wing UAV model Skywalker 1800 used for aerial survey	15
3.3	Flowchart depicting the methodology, yellow shapes represent beginning and finish product, blue intermediate results, green process, red input data	17
3.4	Example of original image and cropped image	18
4.1	Examples of objects in the tiles	22
4.2	Overall accuracy and standard deviation of models	23
4.3	SVM classifier ROC curve	27
4.4	Image classification results	28
4.5	Examples of tiles classified as False Positives	29
4.6	Examples of tiles classified as False Negatives	30
I.1	Image classification results	43
I.2	ROC results for ML models	44

INDEX OF TABLES

4.1	Cropped tiles and labels given based on presence of palms	21
4.2	Accuracy assessment of Classification models	24
4.3	Confusion matrix DT	24
4.4	Confusion matrix GBC	24
4.5	Confusion matrix SVM	25
4.6	Classification results from SVM model for each image	26
I.1	Confusion matrix SVM	45
I.2	Confusion matrix DT	45
I.3	Confusion matrix LR	45
I.4	Confusion matrix GBC	45
I.5	Confusion matrix KNN	46
I.6	Confusion matrix RF	46

INTRODUCTION

1.1 An overview of the work

Agriculture is one of the most important economical activities in the world and specially in developing countries. Nonetheless current agro-industrial means of production carry a heavy impact on the environment, specially palm oil *Elaeis guineensis* plantations. Palm oil has become a very popular crop due it's high demand since it's used in a wide range of products in our society from food, cosmetics, pharmaceutic, bio fuels and its demand has being steadily increasing since 1992 Sumathi et al. (2008). But this high demands also implies land cover changes, and in the case of palm this is happening in an alarming rate Fitzherbert et al. (2008), such that there's international pressure to stop the expansion of new palm tree plantations and effectively manage current croplands. To better manage resources a paradigm shift in the means of production is required, this is a difficult task but with help of technological advancement it can be achieved. Remote sensing techniques are widely applied in diverse fields, but one of the stand out is agriculture Atzberger (2013). Several techniques have being applied with great results with the goal of achieving a better management of resources and reducing the environmental impacts of agriculture. In this context this study will focus on image classification of a palm oil plantation, with the view of helping better manage crop lands by providing information of areas with palms. The method will be based on the use of very high resolution images obtained from an Unmanned Aerial vehicle (UAV), a convolutional neural network (CNN), to extract data from the images and machine learning (ML) algorithms to classify the data.

CNN are fast evolving field and have proven to excel at image recognition task Girshick (2015), Krizhevsky et al. (2012), Long et al. (2015), Ren et al. (2015), and Sermanet et al. (2013), how ever for a CNN to recognize objects it requieres extensive

training, in the sense that it needs to see large amounts of labeled data Krizhevsky et al. (2012). The remote sensing community have taken great interest of the great performance of them in object classification task and have tried to apply them in a remote sensing context. Nevertheless this methods has some drawbacks, as mentioned before they require large amounts of labeled data to learn to detect objects, training one from scratch involves heavy computational burden and time, nonetheless some studies have proven viable methodology to overcome these hurdles. Fine tuning a network is one of the option, this requires knowledge about the architecture of the network, the data that it will be analyzing and several analysis of the data fo obtaining a model that performs the required task, Nogueira et al. (2017) tested this approach obtaining good results. Another option is transfer learning, this method relies on using the capacities of CNN to extract features from image and later using this data coupled with machine learning models to classify or label the data, Sharif Razavian et al. (2014) proved in his study that this "off the shelf CNN" approach yields results that compete with the state of the art methods, this had made these methods popular among remote sensing applications.

Several studies have focused on the use of transfer learning for image classification, but mainly focused on the use of satellite imagery Hu et al. (2015) and Penatti et al. (2015), but not as much in the case of very high resolution imagery obtained from an UAV Zortea et al. (2018). This research can help narrow the gap and serve as reference for how CNN interact with very high resolution images. Based on the previous studies that used transfer learning, good results should be expected, the features extracted with the CNN will be able to differentiate palms in the images.

Therefore this research seeks to determine if features extracted from very high resolution UAV imagery, using a pre trained CNN, can translate to solving an image classification scenario. In this case the classification will be done using ML models and determining which one performs the best for this specific case.

To test this hypothesis, the pre trained `OverFeat` network Sermanet et al. (2013), will be used as feature extractor, and to classify the features six machine learning models will be employed, and their accuracy will be assessed, the best model for this specific case will be used to classify the images of the data set.

This research will provide data about how viable is to use a pre trained CNN and transfer learning approach for image classification, in the context of using a data set consisting of very high resolution images obtained with an UAV. On the other hand this study will provide a method for image classification that can hopefully serve as a base and further developed to produce an efficient resource management for palm tree plantations.

1.2 Aim and Objectives

The main aim of this research is to classify if images from the data set contain palms.

Objectives

- Evaluate if the features extracted by OverFeat transfer well enough to perform classification of very high resolution images.
- Evaluate several machine learning models for classifying features extracted from OverFeat.
- Use the best machine learning model to classify the images in the data set.

1.3 Document organization

The document is organized into five sections. First chapter includes an introduction with an overview about the topic and context in which this research was developed, as well the objectives. Second chapter has theoretical concepts that serve as basis for the completion of this study. Chapter three is a description of the methodology and tools use, chapter four presents and describes results of the study. Final chapter is the discussion of the results and conclusion. Annex I contains results that were not showed on chapter 4 and Annex II contains programming scripts that were used in this study.

LITERATURE REVIEW

This chapter will discuss applications of UAV technologies in the remote sensing field with special interest in previous work linked with image classification using CNN. The second part will about theoretical approaches and application of machine learning models and CNN in remote sensing for image classification.

2.1 Remote sensing and UAV applications

2.1.1 Remote sensing

In a general context remote sensing is the act of acquiring data from an object, phenomena or geographical space with out direct interactionLillesand et al. (2014). The year 1972 is a benchmark for civilian remote sensing with the launch of Landsat 1, this system set the basis and norms for multi spectral sensor technologies Blaschke et al. (2014). From this point on the field has been in a constant state of evolution pushed by technological progress and a growing body of research Atzberger (2013), Colomina and Molina (2014), Navalgund et al. (2007), and Pajares (2015), to the point of becoming a staple in many and diverse applications Basse et al. (2016), Hedley et al. (2016), Masini et al. (2018), Megahed et al. (2015), and Richter et al. (2016). The success of remote sensing allowed for new technologies to be incorporated, such is the case of unmanned aerial vehicles (UAV).

There are many definitions for UAV, for example, the international civil aviation organization of the United nations (ICAO), defines it as: A set of configurable elements consisting of a remotely-piloted aircraft, its associated remote pilot station(s), the required command and control links and any other system elements as may be required, at any point during flight operation."International Civil Aviation Organization (2011). This definition is broad and can encompass a wide range of system, from military grade

equipment to small radio control aircraft, therefore this review will focus on the definition proposed by Watts et al. (2012), as it fits into the type of system that was used for collecting the data for this study. Watts proposed definition considers three main parameters: altitude, endurance and flight capabilities, depending on these characteristics a UAV can fit into an specific category, such as Low Altitude Short-Endurance (LASE) systems. These platforms typically have a payload in the range of 2-5 kg, with a wingspan smaller than 3 meters, up to 2 hours of flight time and can operate in a range of 5 km with a ground station. In other words, this would be considered a "small UAV", and are the systems most commonly used and widespread in research projects Zecha et al. (2013).

Several factors can be attributed to the UAV thriving in remote sensing, technology has played a prominent role with breakthrough in positional and navigation systems Zecha et al. (2013), miniaturization of hardware and sensors Casagrande et al. (2018) and Salamí et al. (2014), that made it technically and economically feasible to create systems capable of autonomously following a flight path and gather remote sensed data. This trend also spread to data processing hardware and software, extracting information became efficient, relatively cheap and accessible to the general public.

Other important factors to be considered are the flexibility a UAV provides regarding temporal, spatial and spectral resolution. The modular nature of these systems allows for seamlessly interchangeable sensors Zecha et al. (2013), as well as the possibility for selecting the best condition and time for surveying an area Salamí et al. (2014). This characteristics and flexibility allows this systems to serves as a complement or even substitute to traditional platforms like space borne or piloted air crafts Salamí et al. (2014). In some cases UAV is the only viable option for accessing or obtaining information of remote and dangerous areas for humans Bollard-Breen et al. (2015) and Everaerts (2008) and finally, depending on the context and scale of the application, they can be a cheaper alternative to traditional methods Iizuka et al. (2018) and Watts et al. (2012).

In conclusion technological advancement made it so, that it's no longer necessary to have highly trained personnel and equipment to build a UAV, obtain data, process it and extract useful information.

2.1.2 UAV applications

There are many applications for UAV data in remote sensing, cartography production is one of the principle examples, Berteška and Ruzgienė (2013) obtained images and produced digital elevations models (DEM), that fulfill the requirements of topographical and GIS applications. Researched done by Mesas-Carrascosa et al. (2014) assessed the quality of orthophotos generated by UAV, the spatial accuracy was evaluated using techniques employed by mapping agencies, with positive results. A new field where UAVs are gaining traction is air quality measurement, as the review by

Villa et al. (2016) shows a growing interest and applications for measuring air quality, atmospheric composition and air pollutant monitoring.

Geomorphology survey are also being done with the help of UAVs, as proved by Wang et al. (2016), with soil erosion monitoring in Sancha river basin in Northeast China. Eltner et al. (2016) presents an overview in the use of high resolution UAV images for recreating physical objects, this methods have being used in the reconstruction of 3 dimensional objects in various context and research projects.

Vegetation studies have been one of the prominent focus of remote sensing, be it for agricultural or forestry applications. In the field of forest management Mikita et al. (2016) used aerial and ground photogrammetry for estimating height and diameter at breast height(DBH), Yuan and Hu (2016) applied random forest models and object-based classification for identifying pest and monitoring health of forest.

Very high resolution data can be used for single weed mapping, for site specific plant protection in wheat fields Pflanz et al. (2018), Luna and Lobo (2016) identified and mapped gaps in sugarcane plantations for optimizing replanting methods, Gago et al. (2015) analyzed the benefits and gaps of water stress monitoring.

Most of the applications listed above rely on identifying objects, defining their features and boundaries. Many methods have being developed for this purpose, with most of them focused on automating this task. In early stages the focus was on identifying large objects do to resolution restrains of sensors. Most of the methods develop where based on statistical analysis of pixels but this meant that underlying spatial patterns where not considered Casagrande et al. (2018).

With the increasing quality of sensors and the widespread use of UAVs identifying smaller objects became feasible, the availability of new data paved the way for a paradigm shift. The core analysis based on statistics were complemented with object classification, this led to the creation of new concepts, relationships between the data and methods of analysis. For this concepts Hay and Castilla (2008) proposed the term geographic object-based image analysis (GEOBIA), they considered to be a a sub-discipline of geographic information science, that focus is creating automated methods for segmenting remotely sensed images into meaningful objects and analyzing their spatial, spectral and temporal characteristics, to create new geographic information in GIS-ready format Hay and Castilla (2008). This field has benefited from the development and advancement of machine learning methods and specially convolutional neural networks (CNN), due to their capacity to solve complex image classification problems Chen et al. (2018).

2.2 Machine learning techniques in Remote Sensing

Machine learning (ML) techniques are approaches to solving classification and regression problems, they are considered "universal approximators", they can learn the inner workings of a system from a large set of data without prior knowledge Lary et al.

(2016). Being able to model complex systems, capacity of using a wide range of input and not assuming normal distribution of the data are some of the main reason they have become popular. These characteristics are also common among remote sensed data Maxwell et al. (2018), therefore this methods have being widely used and applied by researchers and specialist in the field. Some of the most common ML algorithms are decision trees (DT), sel-organizing map (SOM), random forests (RF), support vector machines (SVM) and artificial neural networks (ANN), the latter two have being widely applied in the field of geoscience Lary (2010).

Selecting the best algorithm can be a challenging task, as proven by the comparison done by Maxwell et al. (2018), in which there's no clear algorithm that constantly outperforms the rest. This might be caused by deferring methodological approaches and specific characteristics of the data, therefore the best algorithm is case specific and thus it's recommended that the analysis should include multiple classifiers Lawrence and Moran (2015).

The scientific literature regarding the use of these methods has being steadily increasing, Belgiu and Drăguț (2016) provides and overview of studies using RF classifier, Mountrakis et al. (2011) presents a similar review but for SVM and Maxwell et al. (2018) compares the accuracy of algorithms in a remote sensing context. Recently interest and innovation has gravitated towards the development for deep neural network, a form of artificial neural network, that outperform traditional methods Zhang et al. (2016).

Figure 2.2 exemplifies a basic artificial neural network architecture, it has three neurons in the input layer, that feed neurons into a hidden layer, which in terms activate the output layer. Deep neural networks are several neural networks stacked together, researches found that adding more layers to the network would increase the accuracy of the model. This has being proven in several studies where convolutional neural networks perform image classification task, section 2.3 will focus on this models and their use in remote sensing context.

2.3 Convolutional neural-networks

The first instance of a convolutional neural network is usually credited to the research paper *Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position* Fukushima and Miyake (1982), this was a system designed for visual patter recognition and was inspired on the findings of Hubel and Wiesel (1962), on how a cat's visual cortex is structured and how it works to identify objects. LeCun et al. (1998) method for recognizing hand written digits was also an early application of CNN and served as basis for other research and models.

The year 2012 is a benchmark year, as Krizhesvky, Sutskever and Hitton published ImageNet Classification with Deep Convolutional Neural Networks Krizhevsky et al.

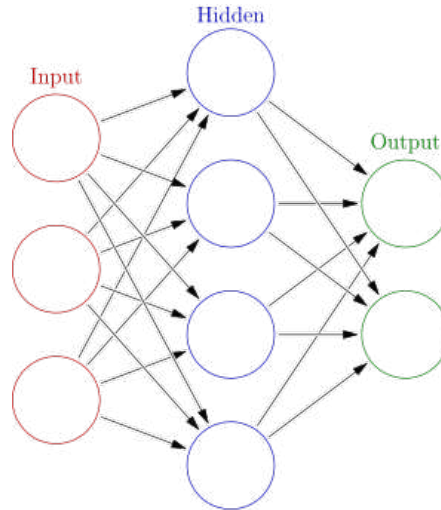


Figure 2.1: Artificial neural network architecture

Image Source: Glosser.ca (2013)

(2012). Considered to be a seminal work in the field of deep learning, it was the first instance that a CNN was used for winning the LSVRC (ImageNet Large-Scale Visual Recognition Challenge), a yearly competition in the computer vision field, which requires the competitors to develop algorithms that can automatically annotated images from a data set Russakovsky et al. (2015). The model proposed by Krizhevsky et al. (2012) called AlexNet, had the lowest error rate in the history of the competition until that point. The authors explained that the limitations in their model was due to technological constrains as well as amount of data available, and predicted that with more data and technological advancement this type of models would increase in accuracy.

From that point on CNN, were a mainstay in computer vision competitions and their popularity started to permeate into other fields. In the coming years modifications where made to the original model, from fine tuning it's parameters Zeiler and Fergus (2013), to simplifying the structure or architecture Simonyan and Zisserman (2014), to creating new and complex model Szegedy et al. (2015). A tendency was established, each new iteration of the CNN model surpassed its predecessor in accuracy.

The fully convolutional network work Long et al. (2015) was another important benchmark in semantic segmentation, it achieved the goal of assigning each pixel in an image to a class. Parallel to the adjustment made to the CNN architecture and accuracy, other steps were taken into making the process faster and adding features, such as defining the boundaries of objects Girshick et al. (2014), this method combined image classification and detection and started a new trend in the field. Subsequent improvements were made to this approach, in its speed and accuracy Girshick (2015) and simplifying its structure Ren et al. (2015).

As these methods matured and proved successful, they became the model of choice

for many applications, including remote sensing. Nevertheless, remote sensed data in general presents some particularities that need to be considered:

- At its core remote sensed data has an important geospatial component that needs to be considered, harnessing these characteristics can lead to better analysis.
- In general, remotely sensed data comes from a wide range of sensors, with different spectral and spatial resolution, as well as different types of data.
- The large amount of data produced with high temporal resolution is creating a shift towards analysis of time series instead of single scene.
- Remote sensing has a traditional approach of detecting and quantifying a phenomenon, using models and expert knowledge. On the other hand, deep learning approaches focus on fully automated expert-free knowledge solutions.

2.4 Application of CNN in Remote Sensing

Despite this context remote sensing scientists have managed to exploit the potential of deep learning for a varying range of applications. Salberg (2015) proposed a method for detecting seal pups, from aerial remotely sensed images. The method uses a pre-trained deep convolutional neural network for feature extraction and support vector machine for classification. Results show the method was successful and implies that the methodology can be generalized and applied to object recognition in different scenarios, but requires further improvements and fine tuning. Chen et al. (2017), used deep learning techniques to count fruits in a tree, first identifying potential regions of interest (the location of the fruits), then comparing it to ground truth data and finally applying linear regression for estimating a final count, the authors propose using this methodology for plant phenotype identification, counting plants and monitor plant disease that present visual symptoms. Hung et al. (2014), proposed a method for obtaining the best flight altitude for a UAV, for achieving the best classifying performance for detecting weeds in crop fields, Zhu et al. (2017) provides a review of applications of CNN, showcasing state of the art methods.

A specific deep learning workflow for detecting and counting palm trees was proposed by Li et al. (2017) their method was tested against manually labeled trees, showing good results, a limitation they encountered, was the size of their data set. Other methods tried to generalize the classification process for different forest types, using a cascade neural network, it showed promising results, but the authors intend to better their method applying a convolutional neural network architecture Tianyang et al. (2018).

2.5 Fundamentals of CNN

A CNN, is a machine learning algorithm, that can learn by analyzing large quantities of data. Its origin stems from artificial neural network models, which are mathematical approximation to how a human brain works. Their general process includes analyzing inputs through a “neuron” or a layer of mathematical operations and classifying the output depending on its value. This process is achieved by an activation function, this function serves as a threshold for defining the class of the output. For image recognition tasks the input are pixel values.

This model's particularity is good for image analysis, because they can deal with the large amount of data by reducing their dimension, thus easing the processing burden. They have proven to excel at extracting features from images, performing classification Krizhevsky et al. (2012) and Szegedy et al. (2015), semantic segmentation Long et al. (2015) and object detection Girshick et al. (2014), Ren et al. (2015), and Zhu et al. (2017).

CNN follow a basic structure and its main components can be classified depending on their functions, next a description of the main components is described. This section is based on the lecture notes from Karpathy (2016).

- **Convolution layer:** This layer receives the input as an array of pixels, each one has three values associated with them, they are width, height and depth, which relates to the number of bands that the image has. This layer transforms all this information into a function that will describe the pixel, in mathematical terms this is called convolution, thus the name. This operation is performed on the whole image by sliding a filter or kernel. The outputs are values from that summarize each pixel.
- **Pooling layer:** this part is responsible for reducing the dimension of the convolved layer, one of the most common methods of doing this is called max pooling, which consists on returning the maximum value from each of the convolution operations within a defined window.
- **Activation layer:** They are in charge of defining which neurons or layers from the network should be activated.
- **Fully connected layer:** is the layer that is in charge of classifying the features extracted from the previous layers and generates an output. As the name suggests this layer connects all the input layers to the outputs.

In summary when an image is fed to a CNN, the convolutional and pooling layers transform the pixel data, this phase is also called feature extraction, this is when the CNN will recognize geometries in an image, for example identify the structure of a palm tree or a dirt road. The classification of the image is done by the fully connected

layers, their output is usually a probability score of what the image is. Figure 2.2 shows the architecture of Alexnet Krizhevsky et al. (2012), its structure serve as foundation to many other models Sermanet et al. (2013) and Zeiler and Fergus (2014).

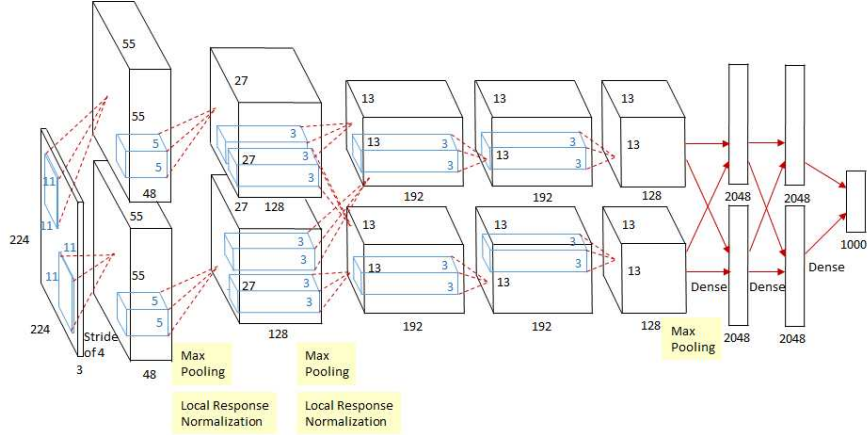


Figure 2.2: Architecture of a Convolutional Neural Network

Image Source: Krizhevsky et al. (2012)

Despite excelling in image classification tasks, these models have some drawbacks such as the need for large data sets of labeled images as well the time it takes to train a model from scratch, nonetheless there are experience of using pre trained models as feature extractor with positive results Marmanis et al. (2016), Nogueira et al. (2017), and Othman et al. (2016). This approach is called transfer learning, it relies on using a pre trained CNN and removing the fully connected layer (layer that provide classification results), and extracting features using the rest of the architecture Karpathy (2016). This method provides the mathematical description of geometries such as edges, curves, size, shapes and angles found in the image. This description are the features from the image and they can be used to train a liner classifier like SVM Zhang et al. (2015). This approach was also applied in this study, using OverFeat Sermanet et al. (2013), which is a pre trained CNN that was used as a feature extractor, this features were later used to train several machine learning classifiers.

2.6 OverFeat

The OverFeat network is based on Alexnet Krizhevsky et al. (2012) with some minor modifications. It has 6 convolutional layers, they have a varying number of neurons from 96 to 1024, the kernel in these layers vary in size from 3x3 to 7x7 and a max pooling layer with kernel size ranging from 3x3 and 5x5, rounded up by 3 fully connected layers. This model introduced a novel way of increasing classification accuracy by training a CNN to classify and detect objects simultaneously. This framework achieved the best result in the localization task of the 2013 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and 4th place in the classification task. OverFeat was trained

using ImageNet 2012 training set, which has over a million images spread out in 1000 classes Deng et al. (2009). This data set contains images that are mainly centered and free from image occlusion and clutter Sharif Razavian et al. (2014).

The best iteration of this model served as base for, `OverFeat` feature extractor, that was used in this study. There are several instances of the use of `OverFeat`, Nogueira et al. (2017) compared three different approaches to using CNN in remote sensing. In the study they built and train from scratch a CNN, fine tune a pre-trained CNN and use `OverFeat` to extract features and classify images. Marmanis et al. (2016) proposed a two step classification process where they used the pre-trained `OverFeat` model coupled with another CNN in charge of classifying the features extracted. Sharif Razavian et al. (2014) used `OverFeat` for various image recognition task, with different data set obtaining results that compete with highly sophisticated and highly tuned state of the art methods.

In this study the `OverFeat` network was used as it has yield good results for classification task, also the author provide a version of this model as a ready to use software (<https://github.com/sermanet/OverFeat>) that can perform feature extraction on our data set with out the need of high computational cost.

DATA AND METHODS

This section will describe the context of the study area, how the data was gathered and the most important characteristics.

3.1 Description of study area

The data was gathered on September 27 2014, near Loma de Mico village 12°11' 45.54" N, 83°49' 48.04"W in the Municipality of Kukra Hill, which is part of the Autonomous region of the Caribbean south coast (RACCs) of Nicaragua ([Figure 3.1](#)). Loma de Mico is considered to have a tropical monsoon weather, wet season extends for 10 months, this translates to 2000 - 3000 mm of rain fall. Average temperature ranges from 24 - 27 °C. Rolling hills with slopes between 20 - 30% are staples of the relief, with the highest point rising 192 meter above sea level. The aerial survey was done over a palm tree (*Elaeis guineensis*) plantation with an area of 45 ha. In it several types of land cover could be distinguished, most prominently present were mature palm trees, palm trees in developing stages, dirt roads, patches of natural forest, grasslands and scatter infrastructure.

3.2 UAV System

The system used for the aerial survey was a fixed wing glider UAV ([Figure 3.2](#)). The airplane is a skywalker model 1800 from 2014, which is originally manufactured as a remote control plane with focus on providing the users first person view (FPV) capabilities. The RC airplane was modified to fit larger payloads (camera, batteries) and with an autopilot, sensors and navigation system making it capable of autonomous

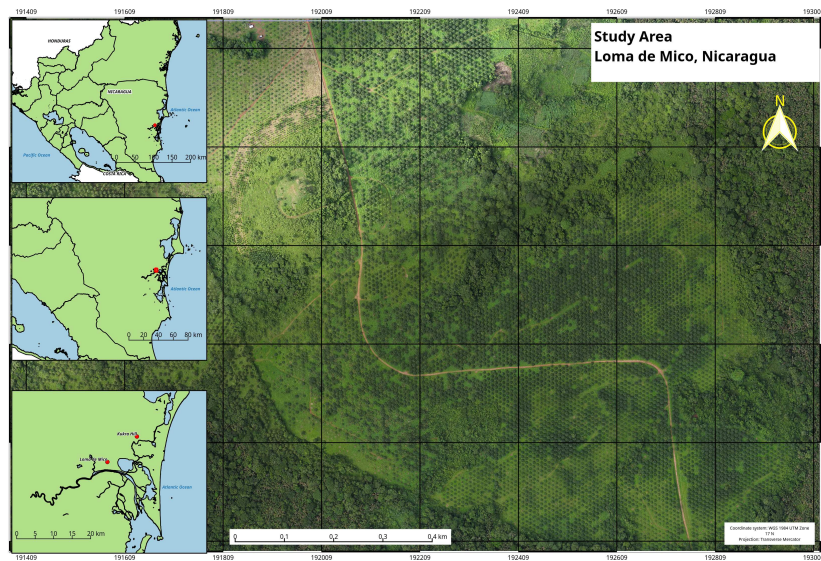


Figure 3.1: Study area in Loma de Mico, East Nicaragua

flight, this was done by the consulting firm [EVOLO](#). Next a detailed description of the model is given.

The air frame or body was made from EPO (Expanded PolyOlefin) foam, had a total weight of 5 kg, a wingspan of 1800 mm and an endurance of 35 minutes at a speed of 40 km/h. The system used an ArduPilot Mega autopilot APM 2.5, capable of autonomous stabilization and way point navigation, coupled with compass, GPS, barometric pressure sensor and telemetry communication, the system as a whole is capable of completing a pre loaded flight plan autonomously.

The ground control station was a laptop running Mission Planner software version 1.3.10 connected to a radio receiver, this allowed real time communication with the UAV, during the mission it transmitted position (X,Y), height above ground level, yaw, pitch, roll angles, speed and battery levels.

Images were collected using a Canon PowerShot S100 camera (Canon Inc., Tokyo, Japan) this model has a complementary metal-oxide semiconductor (CMOS) sensor size 7.53 x 5.64 mm and an approximate area of 42.5 mm² which fits 12,100,000 pixels. Custom firmware CHDK version 1.2 ([Canon Hack Development Kit](#)) was setup on the camera. This free software was developed by Canon PoweShot users with the purpose of adding features and enhancing the capabilities of the "point and shoot" camera. In this study it was used to program the camera to take pictures every second by increasing the shutter speed, this would not be possible using the default software provided by the manufacturer.



Figure 3.2: Fixed wing UAV model Skywalker 1800 used for aerial survey

3.3 Aerial survey

The aerial survey started at 15:00 and ended at 15:07 for a total of seven minutes, we assume that atmospheric conditions are uniform in all the images, because during the survey no extreme climatic variance was observed. The flight plan was designed using Mission Planner, taking into consideration characteristics of the camera as well climatic conditions, altitude was set at 280 m above ground level, cruise speed of 40 km/h, ensuring that there was a 80% overlap between the images.

3.4 Data set

The aerial survey produced a surplus of 500 images with a resolution of 10 cm, despite this ample amount of images most of them had issues like image blur or grainy quality making it difficult to distinguish any objects, therefore pictures with good quality were selected, this mounted to a total of 11 images. Factors considered in the selection were, representation of the objects present in the study area, quality of the image, for example how easy it is to distinguish the objects, and having a balance between the images with palms and no palms. To better describe the images percents of palm per image as calculated, this was done after the cropping process described in [section 3.6](#). The image is subdivided into 192 tiles, the tiles that contain palms are counted and based on the amount the percentage of palms in the images is obtained. This value is referenced in the description of each image.

To extract features from the images using OverFeat they had to be cropped into tiles of 250 x 250 pixels. Each image produced 192 tiles, the ones containing palms were counted and used to estimate the percentage of palms in one image. For example image 2389, has 112 tiles containing palms, which represents 60% of the whole image. Next is a description of the images selected:

- IMG_2361: Mature palm trees are the most prominent objects in the image, it also features a small patch of natural forest, two parallel dirt rows that transverse the image and patches of grassland. The palm trees distribution is systematic but patches of missing palm are noticeable in the image. Palm trees represent around 92% of the image.
- IMG_2367: Contains three plots of palm tree saplings, a small patch of mature trees and a section of natural forest surrounded by two plots of grassland. The mature and saplings trees are all in a symmetrical distribution and represent 60% of the image.
- IMG_2371: The image is divided between a plot with mature palms and small ones that are too small to effectively distinguish, a short dirt road a small patch of grassland and natural forest complete the image. Palms are distributed in a symmetrical manner, only the mature palms were considered for the analysis and they represent approximately 59%.
- IMG_2374: Natural forest is the most prominent land cover in the image, it also contains two patches of mature palm symmetrically spaced, a short dirt road and a grassland. The palm trees represent 33% of the image.
- IMG_2376: Mainly consists of natural forest and the some sparse mature palm trees, they represent 22% of the image.
- IMG_2377: Comprised of natural forest with randomly sparse mature palm trees mixed with the forest, the palms represent less than 1%.
- IMG_2378: Natural forest represents the majority of the image, with a plot of mature palms symmetrically spaced and a short dirt road. Palm trees represent 22% of the total image.
- IMG_2389: Image presents a small patch of natural forest mixed with some randomly spaced mature palms, a lot with palm tree saplings, grassland plot, two dirt roads and small housing infrastructure. The evenly distributed palm tree saplings represent 58% of the image.
- IMG_2392: The image contains saplings and mature palm trees symmetrically distributed, a very small patch of natural forest, a main dirt road with 4 secondary roads and some scattered housing infrastructure. Palm trees represent 97% of the image.
- IMG_2395: Evenly spaced mature palms make up most of the image, there's a small patch of natural forest a main road that branches into two secondary roads the palm trees represent 95% of the image.

- IMG_2402: A patch of natural forest surround a plot of mature palm trees and some grassland area, the palm tree represent 46% of the image.

3.5 Methodology

The general approach for the project is described in this section. The main goal was to detect if an image had palm trees, using a pre-trained convolutional neural network called *OverFeat*. This CNN architecture is not particularly trained to classify palm tree, nonetheless it excels at extracting features from images, meaning that it can describe in a mathematical context the geometries (objects) in an image. Cropping our data was part of the pre process, this was done to comply with data requirements from *OverFeat*, subsequently they were fed to the CNN for feature extraction and then classified using machine learning models. After assessing the accuracy, the best model was picked and used for visualization of results. A simplified methodological flowchart is present in Figure [Figure 3.3](#), yellow shapes represent the input data and the final product, blue are intermediate results, green represent process and red input data for the process.

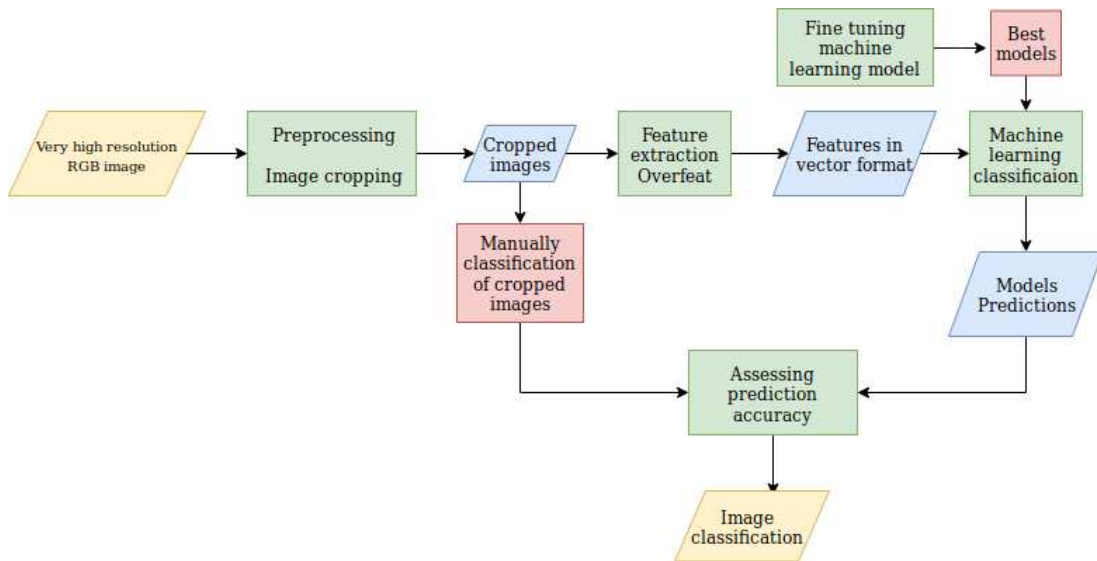


Figure 3.3: Flowchart depicting the methodology, yellow shapes represent beginning and finish product, blue intermediate results, green process, red input data

3.5.1 Tools

In this section the tools used are listed and briefly described, all the codes used are in [Annex II](#)

- ImageMagick version 6.9 was used for cropping the images.
- *OverFeat* was used as a feature extractor.

- Python 3.6.5, numpy to load and manipulate the data set, scikit-learn was used for building and fine tuning the machine learning models, as well as making the predictions and matplotlib for plotting results.
- LibreOffice Calc was used for labeling the images, as well for assessing the accuracy of the models
- QGIS 3.0 was used for creating the map of the study area, and for visualization of the results.
- SQLite for managing all the files and tables created
- All processes were done using a laptop with an Intel i7-6700HQ CPU @ 2.60GHz and 8 GB of DDR 3 RAM

3.6 Data Preprocessing

The purpose of this step is to generate images that can be analyzed with OverFeat. The original images have a size 4000 x 3000, each image was cropped into 192 tiles of 250 x 250, generating a total of 2112 images, the size was selected based on the requirements of OverFeat. In [Figure 3.4](#) shows an example of image 2392 previously to being pre processed (left) and an example of a tile from the cropped image (right). ImageMagick software was used for the cropping process, this is a free open source software for displaying and manipulating images. To automate this process a bash ¹ script was created. It works by accessing the directory where the images are located it crops each image into 192 tiles, giving each tile an specific name and exports the results into a new directory, it repeats this process for each image. The output was classified depending on the presence of palms. The classification was performed via visual inspection and relying on the experience and knowledge of the area obtained during the aerial survey, nomenclature used is as followed, images containing palms were labeled as "1", otherwise a "0" was assigned.



Figure 3.4: Example of original image and cropped image

¹UNIX based programming language

3.7 OverFeat and Feature Extraction

OverFeat software with its pre trained parameters was obtained from the creators [Github page](#), this software was made publicly available and provides a convolutional neural network based image classifier and feature extractor, in this study only the feature extractor was used. Again a bash script was used to automate the process, the script takes as input cropped images produced on [section 3.6](#) and feeds to OverFeat, feature extractor. The output is a file that contains the mathematical description of each shape found in each image (see [Annex I](#)), the description has a magnitude and direction associated, therefore it can be considered a vector, that's to say each feature extracted is represented as a vector.

These vectors were linked to the file containing labels for each image, producing a file that has image identification name, labels and a mathematical vector describing objects in the image, this file was used to train the machine learning models. It's expected that the features extracted from images labeled as 0 and 1 vary substantially, making it possible to train a machine learning model to predict the label of the features extracted. this file was used to classify the vectors with machine learning models.

3.8 Machine Learning Models

Models were built using the python Scikit-learn library, a very common and wide spread tool in machine learning applications and scientific field. Many factors make it popular such as allowing the user to developed well known machine learning algorithms with a easy to use interface, it has a Berkley Software Distribution (BSD) license which impose minimal restrictions to its use and distribution, minimum requirements for ruing, ample documentation and examples, designed to maximize computational efficiency and ease of use Pedregosa et al. [2011](#).

Taking advantage of the capabilities and features of this library, several models were built they were:

- Support Vector Machine Linear
- K Nearest Neighbor
- Decision Tree
- Random Forest
- Gradient Boost
- Logistic Regression

The models are python scripts that use as input the file containing labels and vectors (generated in [section 3.7](#)). The script splits the data set into 70% for training

and 30% testing. The training data set was used for fine tuning the models, best parameters were selected and used to build the final model. The final product was a machine learning model that took as input a file containing vectors with no labels and produced a classification result for each vector. Finally to summarize the results a file was created with the classification provided by each model and the labels manually given. Example scripts for the models and fine tuning process are in [Annex II](#)

3.9 Visualization

To aid with the visualization of results a shape file for each image was created, the file contained name of each tile and classification result from each model as well as the manually given label, this allowed to compare between the classification result of each model.

3.10 Accuracy assessment

The predictions made by the models were compared with the labels given manually, for estimating the model accuracy. Overall accuracy was estimated using the formula:

$$OA = \frac{TP + TN}{TP + TN + FP + FN}$$

Where True Positive (TP), represent the positive cases correctly classified, True Negative (TN) the negative cases correctly classified, False Positive (FP), the negative cases classified as positives and False Negative (FN), the positive cases classified as negative.

RESULTS

4.1 Result overview

Results are presented in this chapter, very high resolution images were used as input for extracting features using a pre trained CNN, the features were used to train machine learning models and to classify the images, the best model (SVM) achieved a accuracy of 97%.

4.2 Image preprocessing

The cropping process for each image produced 192 tiles, generating a total of 2112 tiles. Each tile was given a name based on the number of the image and the number of tile. The list with names was used for labeling each tile depending on the presence or absence of palms, in total tiles labeled as "1" were 1001, meaning tiles with palms. The remaining 1111 were labeled as "0" meaning no palms. [Table 4.1](#) provides an example of the name given to a tile, the image it belongs to, the number of the tile under "crop" column and the label given. This table contains six records as an example of the results from the cropping process and the nomenclature used for each tile.

name	number	crop	label
IMG_2361_250x250_000.jpg	2631	0	1
IMG_2367_250x250_100.jpg	2367	100	0
IMG_2371_250x250_021.jpg	2371	21	1
IMG_2374_250x250_029.jpg	2374	29	0
IMG_2376_250x250_031.jpg	2376	31	0
IMG_2392_250x250_020.jpg	2392	20	1

Table 4.1: Cropped tiles and labels given based on presence of palms

A detailed look of the objects present in the tiles is presented in [Figure 4.1](#), it shows examples of the most representative objects of the study area, such as palm in development stages and mature, dirt road, forest, human made infrastructure and grassland. Items (a), (b) and (c) are tiles classified as 1, on the other hand items (d), (e) and (f) are classified as 0.

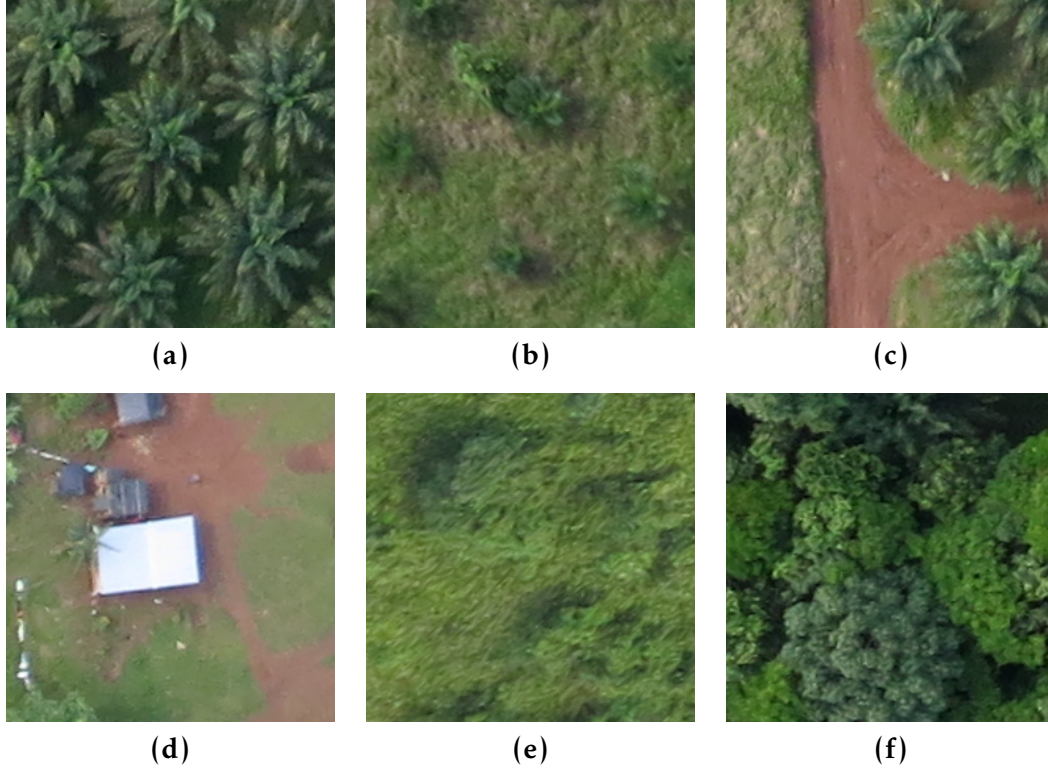


Figure 4.1: Examples of objects in the tiles

(a) Mature Palms (b) Palms in development stages (c) Dirt road (d) Infrastructure (e) Grassland (f) Forest

4.3 Feature extraction with OverFeat

The cropped images were processed with `OverFeat`, the output of this process was a file containing features from each tile. This is a vector of 4096 dimensions also called CNN codes Karpathy (2016), specifically its a text file containing a list of values that represent in a mathematical form the geometries found in the tiles. This data was merged with the file generated in [section 4.2](#) and made it possible to relate labels to the extracted features file. Examples of the extracted features for tiles labeled as "1" and "0" can be found in [Annex I](#).

4.4 Machine learning models classification results

The data set generate in [section 4.3](#) was used as input for training machine learning models, the data was randomly split 70% for training and 30% testing, the models were fine tuned and the best parameters were selected and then tested. [Figure 4.2](#) summarizes the overall accuracy for each model, DT model achieved the lowest accuracy with $86.88\% \pm 1.11\%$, the second lowest was RF model with $90.50\% \pm 0.71\%$, the next three models had similar performance, GBC $92.66\% \pm 0.79\%$, KNN $94.18\% \pm 0.63\%$ and LR $95.88\% \pm 0.60\%$, finally the SVM model had the highest accuracy with $97.73\% \pm 0.51\%$.

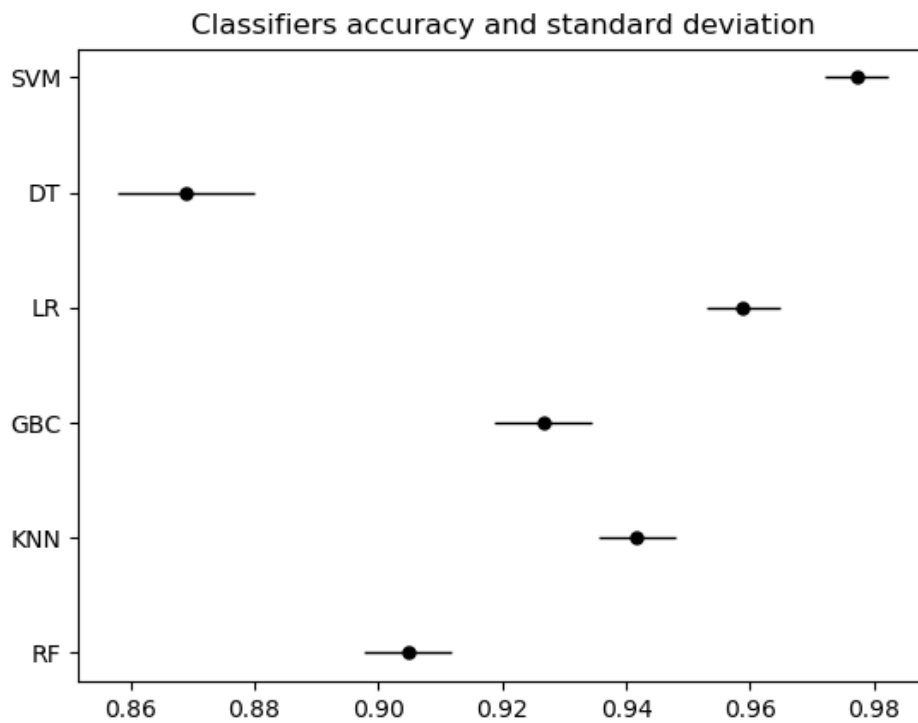


Figure 4.2: Overall accuracy and standard deviation of models

[Table 4.2](#) provides a similar description of the performance from the classifiers, but also includes values of sensitivity, specificity and precision. With regards to sensitivity the top three values follows the same order as observed in the overall accuracy, the forth highest value correspond to the RF model followed by GBC, this is the inverse order compared to overall accuracy. The lowest value corresponds to DT model, as is the case with overall accuracy. Values for specificity follow the same pattern as the overall accuracy, while the precision mirrors the order found in the sensitivity. Generally there's a tendency that the DT model provides the lowest values regardless if is sensitivity, specificity, precision and therefore overall accuracy, random forest

performance better than DT, but not as good as the cluster form by GBC, KNN and LR, these three models have a similar performance but from the three GBC performs worst and LR achieving the best performance. Finally the SVM has the general highest performance from all the models.

Classifier	Sensitivity	Specificity	Precision	OA	STD
SVM	97.30	98.11	97.89	97.73	0.51%
LR	96.30	95.50	96.88	95.88	0.60%
KNN	95.80	92.71	96.38	94.18	0.63%
GBC	93.31	92.08	93.87	92.66	0.79%
RF	94.71	86.68	95.28	90.48	0.71%
DT	85.61	88.03	86.13	86.88	1.11%

Table 4.2: Accuracy assessment of Classification models

Performance of the models are presented in a more tangible matter in [Table 4.3](#), [Table 4.4](#) and [Table 4.5](#). These confusion matrices represent the classification results of DT, GBC and SVM models, these three were compared because they represent the range of performances, the models presents the lowest and highest accuracy (DT, SVM) as well a model which performed in the middle ground between the to extremes (GBC).

The tables have two rows and two columns, the header of the columns are "0" and "1" each representing the labels. The combination of rows and columns produce the values for true positives (TP) "1 1", tiles with palms correctly classified, true negatives (TN) "0 0", tiles with no palms correctly classified, false positives (FP)"0 1", tiles with no palms classified as having palms and "1 0" false negatives (FN), tiles with palms, that were not recognized by the model.

	Reference			Total Result
	DT	0	1	
Predicted	0	978	144	1122
	1	133	857	990
Total Result		1111	1001	2112

Table 4.3: Confusion matrix DT

[Table 4.3](#) presents the results for DT, the model with the lowest performance. Tiles classified as true positives amount 857 and true negatives 978, leaving a total of false positives of 144 and false negatives of 133.

	Reference			Total Result
	GBC	0	1	
Predicted	0	1023	67	1090
	1	88	934	1022
Total Result		1111	1001	2112

Table 4.4: Confusion matrix GBC

[Table 4.4](#) details the results of from gradient boost classifier, it represents the middle ground of performance, it correctly classified 934 tiles with palms (true positives) and 1023 without palms (true negatives). False positives tiles sum a total of 67 and false negatives 88.

	SVM	Reference		Total Result
		0	1	
Predicted 0	0	1090	27	1117
Predicted 1	1	21	974	995
Total Result		1111	1001	2112

Table 4.5: Confusion matrix SVM

[Table 4.5](#) is focus on SVM, it classified correctly 974 from 1001 (True Positives) tiles with palm trees, in the case of tiles with no palm it classified correctly 1090 from 1111 (True Negatives). It classified 21 tiles as false negatives and 27 as false positives.

The results presented in this section points to the SVM as being the best model, considering this tendency the image classification results will focus on the output provided by the SVM model.

4.4.1 SVM classification results

The SVM model used a linear kernel and the cost parameter was set to "C=1000". The linear kernel is recommend for data sets that can be linearly separated, also is less complex compared to the radial base kernel, thus it has less parameters that need fine tuning. The cost parameter was obtained during the fine tuning process. In [Table 4.6](#) a breakdown of the classification results per image, for example images 2376 and 2377 have no false negatives, this is in accordance to the description given in [section 3.4](#), as both this images contain a low percentage of palms, 22% and 1% respectively. Images 2371 and 2378 each have one tile classified as false negative.

The image with the most FN tiles is 2395 with 6 followed by 2389 with 5 and 2367 with 4, in these images palm represents at least 60% of the image, mixed with other features and land cover types such as forest and grasslands, dirt roads

The case with more FP tiles correspond to image 2371 with 5 cases, this image is composed by mature palm plantation, grassland and forest. Images 2376 and 2389 have 4 cases each of FP, this images are contrasting, as the first one is mainly natural forest and palm only represents 22% of the image and the lather is mainly palm tree plantation mix with forest, grassland and infrastructure. Is important to notice that image 2389 presents the most tiles wrongly classified with a total of 11, on the contrary image 2377 has no wrongly classified tiles. These two images are very different from each other, the description provided in [section 3.4](#) shows that 2389 contains, several land cover types, while image 2377 is mainly one land cover type corresponding to natural forest.

image Name	TP	TN	FP	FN
2361	171	19	0	2
2367	85	102	1	4
2371	109	77	5	1
2374	43	145	1	3
2376	9	179	4	0
2377	7	185	0	0
2378	27	162	2	1
2389	118	65	4	5
2392	170	19	0	3
2395	172	13	1	6
2402	63	124	3	2
Total	974	1090	21	27

Table 4.6: Classification results from SVM model for each image

Adding to the performance analysis, [Figure 4.3](#) shows the receiver operating characteristics (ROC) curve. This analysis provides an insight into how well the model separates data into tiles containing palm and no palm. The horizontal axis contains the values of the false positive rate and in the vertical axis the true positive rate, this values are closely linked to the specificity and sensitivity respectively. The curve has an upward trend and levels out, near the top of graph, where the values of true positive rate is close to 1. The curve produces a high value for the metric area under the curve, in this case 95 % of the area of the graph is under the curve, this means that the model is good a separating the classes. As a reference the red dotted line that transverse the graph represents a 50% probability of correctly classifying the data or random chance. The further away the blue curve is from the red dotted line means that the model is better classifier compared to random chance.

4.5 Image Classification

Following the results from [section 4.4](#), classification of the images was done based on the SVM model. Predictions were made on the data extracted in [section 4.3](#), and compared to the labels manually given. A shape file with the classification results was created and used to aid visualization. This was done by stacking the shape file over the image file and categorizing the tiles by color. FP and FN tiles were given magenta and cyan respectively and a wider border to make then stand out. TP and TN negatives were given blue and red colors, less focused was given to this tiles as they were correctly classified.

An example of the classification is presented in [Figure 4.4](#), emphasis is given to this sub set of images as the contain most of miss classified tiles. The complete results for all the images are presented in [Annex I](#). Item (a) is the classification results for image 2367, its a mix of palms and natural forest, it presents 4 cases of FP and 1 of FN. Item

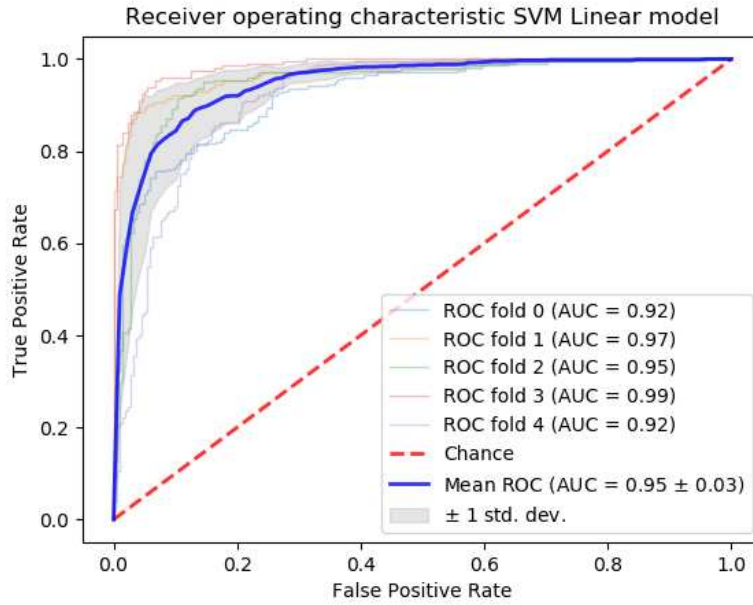
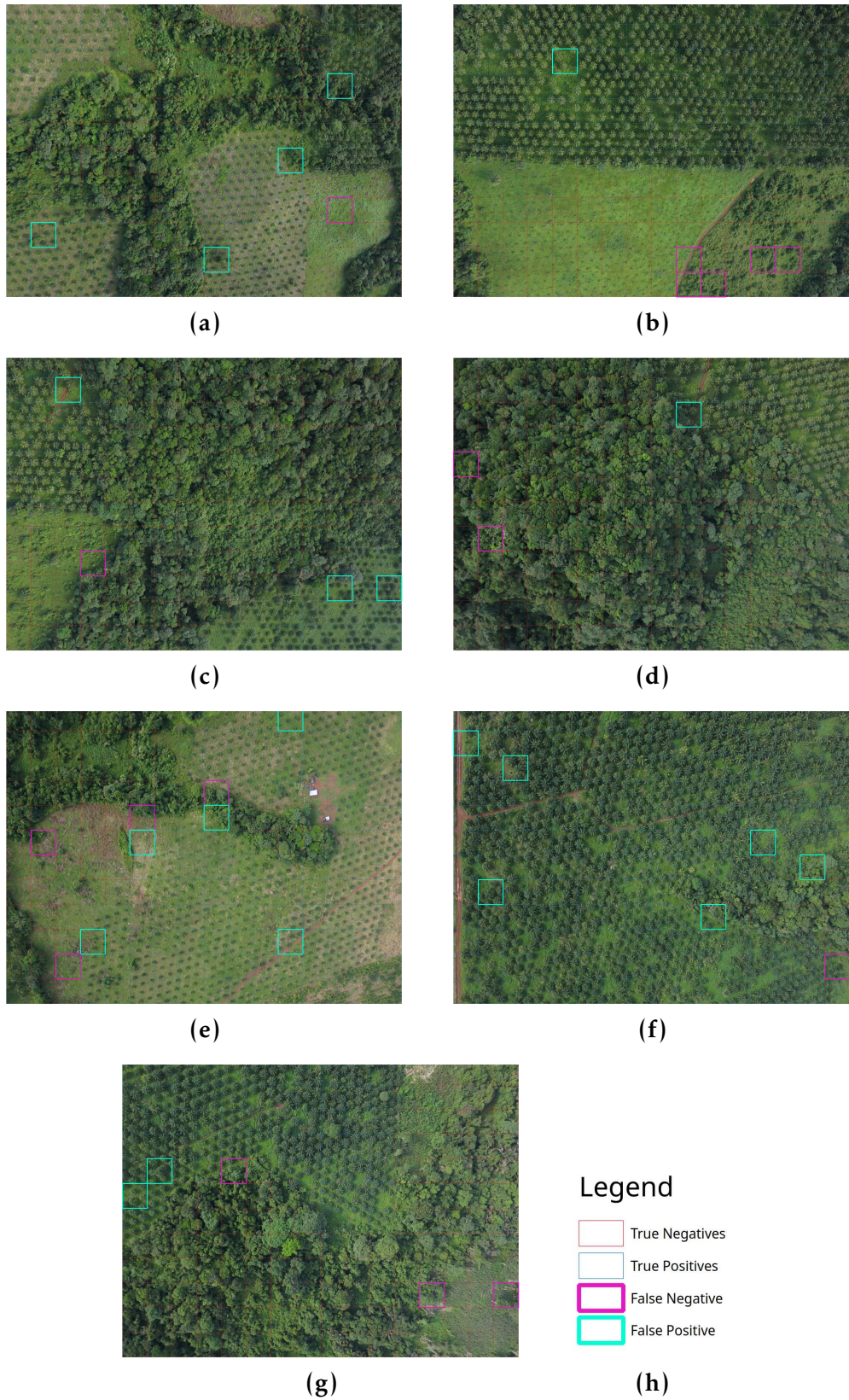


Figure 4.3: SVM classifier ROC curve

(b) corresponds to image 2371 which is a mix between palms and grasslands, it has the most cases of FN tiles from this sub set with 5 and only 1 case of FP. Image 2374 is showed in item (c) is similar in content to item (a) it presents 3 cases of FP and 1 of FN, item (d) is image 2378 is mainly natural forest and a small patch of palms it has 2 cases of FN and 1 FP. Image 2389 corresponds to item (e) it has the most cases of miss classification, 4 FN and 5 FP, the image has a patch of natural forest palms, dirt roads and human made infrastructure. Item (f) is mainly palm trees and a bit of natural forest, it represents image 2395 and has 1 case of FN but the most FP cases from this sub set with 6. Item (g) is image 2402, it has a similar structure to item (a) and has 3 FN and 2 FP. In general features presented in this figure share the characteristics of having several land cover types, meaning that the images contain transitions areas, where is not clearly

Figure 4.5 presents examples of FP tiles, they were given label "0" but the model gave it a incorrect label of "1". In general this tiles can be found in transitions regions, for example item (a) is the transition zone between a patch of forest and palm plantation. Items (b) and (d) are grasslands but that have bush like plants that share resemblance with palm in developing stages, item (c) is forest land cover with several types of trees, it presents a complex structure that confused the classifier.

To have a closer look and analyze tiles classified as FN Figure 4.6 was created. Examples in the figure show palms that were not recognized by the model, item (a) contains mature palms and a dirt road intersection, item (b) is showing some scatter palms not yet fully develop, item (c) are palms near the beginning of a patch of forest and item (e) are several mature palms and one small palm. In general this tiles show



(a) image 2367 (b) image 2371 (c) image 2374 (d) image 2378 (e) image 2389 (f) image 2395 (g) image 2402 (h) Legend

Figure 4.4: Image classification results

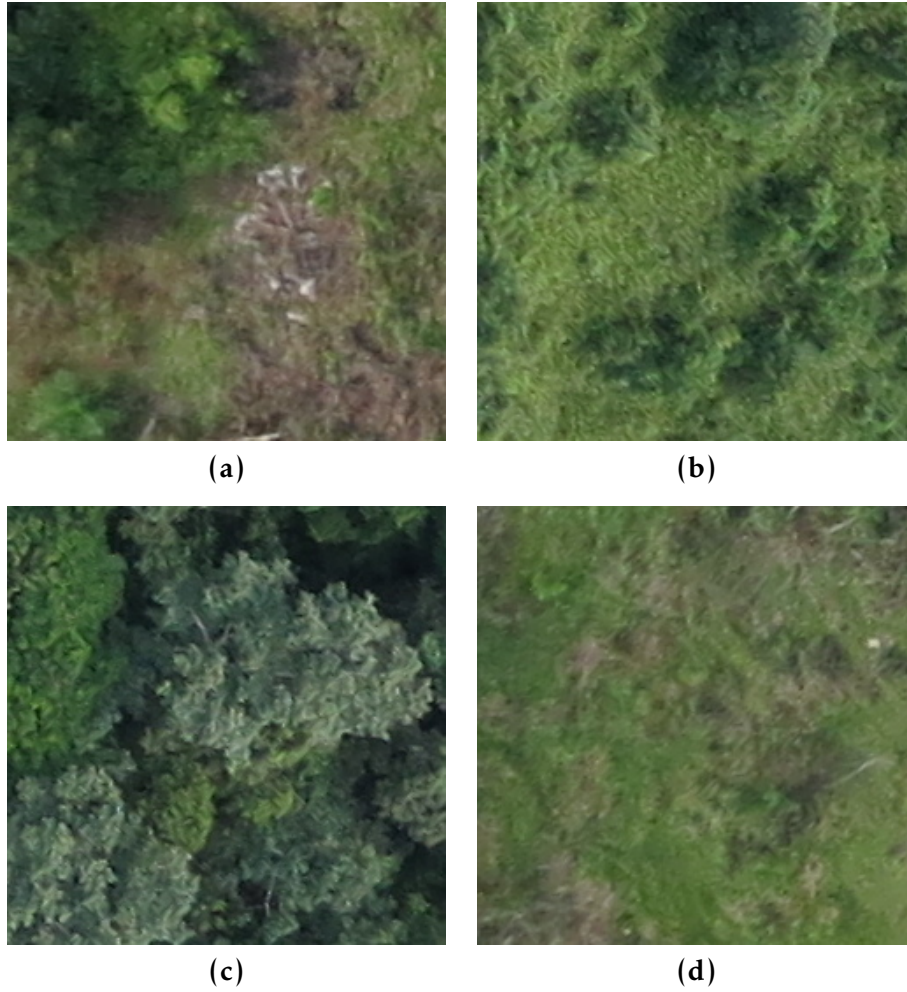


Figure 4.5: Examples of tiles classified as False Positives

(a) image 2389 (b) image 2371 (c) image 2374 (d) image 2389

palms mixed with other objects, close to transition zones, grainy images or with motion blur and palms with different development stages in the same tile.

The results presented in [Figure 4.5](#) and [Figure 4.6](#) paint a general picture of where the model has problems with classification. In general terms this are areas with a lot of different objects such as palms with dirt roads or forest. [Figure 4.5](#) is a close up to examples of images classified as false positives. The figure present some common characteristics as they are low quality images, where detecting palms can be difficult even for humans, also some palms are in developing stages and as such don't have the characteristic mature palm shape, other examples include isolated palms and transitions zones where the land cover changes from plantation to something different. A brief description of the tiles is presented next: (a), image 2389, there's a small bush that resembles a palm, (b) image 2371, grassland with and bushes, (c) image 2374, several trees forming a complex structure and (d) image 2389, the image is blurry and has plants that are difficult to distinguish or confidently classify as palms. [Figure 4.6](#)

share some of the same characteristics, errors were located near transition zones, near the boundaries of the plantation or tiles with different land covers. It was also noted that tiles where there is a mix of palms in different development stages are problematic, palms in developing stage have a tone of color that makes them similar to bushes, thus making them difficult to detect even to the human eye. The tiles presented in the figure belong to (a) image 2392, it has mature palms and a dirt road, (b) image 2389, small palms and grassland, (c) image 2374, small palms next to forest and (d) image 2395, mature palms mixed with small palms.

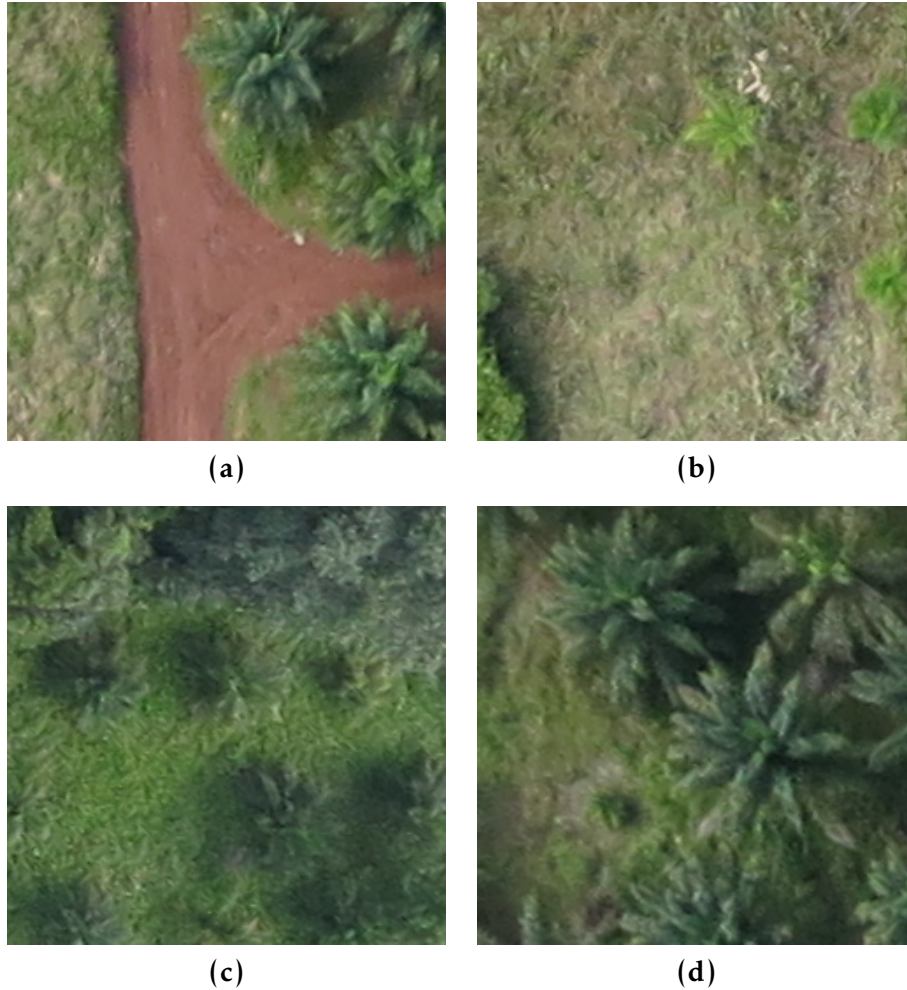


Figure 4.6: Examples of tiles classified as False Negatives

(a) image 2392 (b) image 2389 (c) image 2374 (d) image 2395

DISCUSSION AND CONCLUSION

A discussion of the results is presented in this chapter, in [section 5.1](#) an in depth discussion of each result is presented follow by the main limitations of the study and future works, [section 5.2](#) presents the conclusion of this work

5.1 Discussion

CNN have proven to excel at image classification tasks in a remote sensing context, some cases studies and examples were discussed in [2.4](#). Most of this applications focus on the analysis of satellite images, as this type of data sets are more accessible in comparison to very high resolution UAV imagery. This translates into less research done with very high resolution images and CNN. The method propose in this study tries to narrow this gap by classifying palm tree images obtained from an aerial survey, with the aid of CNN and machine learning algorithms.

The methods was tested by using the pre trained feature extractor `OverFeat`, the output of this network was used for training machine learning models to performance classifying tasks, the results prove that the features extracted are suitable for classifying very high resolution images. In the studies by Nogueira et al. ([2017](#)) and Penatti et al. ([2015](#)) concluded that a trained CNN to recognize everyday objects, can generalize features well enough to be used in a remote sensing context, this aligns with the findings of this study.

The data used for this method consisted of 11 images, these were selected based on their quality and being representative of the study area. The images required a size of 256 x 256 to analyze them with `OverFeat`, this lead to cropping each image and producing a total of 2112 tiles. Small data sets like this are not enough to train from scratch a CNN. In this cases the recommend approach is to use transfer learning

Nogueira et al. (2017) and Sharif Razavian et al. (2014).

As seen in [Figure 4.1](#), [Figure 4.5](#) and [Figure 4.6](#) the cropping process, generates tiles that in most cases contain more than one single palm tree or cases that only partially show palms. Finding a cropping size that complied with the requirements of `OverFeat` and synced perfectly with the spatial distribution of the plantation, so that only one palm was present per tile was not practical. Therefore this method cannot localized or detect the palms in the tile, rather it can be used as a classification method, by labeling if the tile contains palm or not.

In general it can be said that `OverFeat` was able to recognize the difference between tiles with palms and no palms, proof of this is the classification results from the several machine learning models ([Table 4.2](#)), this was possible due to the difference between the features that describe the two classes, as their characteristics were different enough for the ML models to recognize. From the results presented by Nogueira et al. (2017), Zhang et al. (2015), and Zhou et al. (2016), it was expected for the CNN to provide features, that would effectively separate objects in the images. Is important to note that this results were based on satellite imagery, while the results obtained from this study were based on very high resolution UAV imagery. Considering this, it can be said that a pre trained CNN can effectively be used in image recognition task, in the particular case of very high resolution aerial images

In regards to the machine learning models, performance observed coincides with previous studies, in the case of DT, this method is susceptible to data set size, Pal and Mather (2003) showed that accuracy has a positive correlation with training size, this notion was also confirmed by Rogan et al. (2008), this study found that decreasing the training samples implies a reduction in accuracy. On the other hand findings by Foody and Mathur (2004) determined that SVM is less sensible to small sample sized compared to DT. This trend can also be observed in the results ([section 4.4](#)), where DT has the lowest accuracy and SVM the highest, it can inferred that the data set was not big enough for the DT model to have an optimal performance and accuracy, but regardless of the size the SVM still performed well. This helps to explain the reason SVM is one of the most popular ML methods for image classification task, Bazi and Melgani (2018), Petropoulos et al. (2012), Salberg (2015), and Waske et al. (2009).

In the study by Lawrence and Moran (2015), they systematically tested several machine learning methods, with 30 different data sets and found that ensemble methods like random forest performed overall better, nonetheless in some of the data sets SVM performed better. With this findings they concluded the best algorithm for image classification task is dependent of the data set. In this study two ensemble methods were tested, random forest and gradient boost classifier, the first in general performed poorly in comparison to the rest, while the performance of the lather was consider a middle point between all the models. The systematical testing done by Lawrence and Moran (2015), found that assemble methods outperform the rest, in contrast our study shows SVM performs the best, this affirms the conclusion they reached about how the

characteristics of the data set affect the performance of the models. On this grounds selecting a global best model is not possible, nonetheless some reasonable approaches are, trying different models and see which one fits best to and specific data set and revising previous experiments that can give a starting point about which model to select Maxwell et al. (2018).

The overall accuracy of the model is 97.73 %, this is a similar value presented by Salberg (2015). In this study the author used a similar method but used the network proposed by Krizhevsky et al. (2012). The features extracted were also used as an input for an SVM model for classifying seals in aerial images, this method yielded and overall accuracy of 98.2%. The SVM used a linear kernel and the penalization parameter of $C = 10000$. This is a high value for "C", this means that the model will give a high penalization to every wrongly classified tile, this will force the model to search for the solutions with the least amount of miss classifications. The large C value can provide an accurate classifier or in the case that there's noise in the data set it will produce an over fitting classifier Silva (2016).

The model's sensitivity (97.30%) and specificity (98.11%) values are presented in [page 24](#). Sensitivity relates to how well the model detect positive cases (palms) and the specificity measures how well the model recognizes negative cases (no palm). Also the ROC graph [Figure 4.3](#) shows that the area under the curve is 95%, which means that the model does a god job at separating the classes in the data. All this metrics prove that the SVM model is classifying correctly the data, and that the features extracted from `OverFeat` are different between tiles with palm and no palms.

The performance of SVM model can be seeing in the image classification results ([Figure 4.4](#)) the images with least miss classified tiles (2361,2377,2378) are generally images with one prominent land cover, be it forest or palm tree plantation. On contrary the images with the most miss classified tiles (2389,2371,2395) are images with several land cover types, where there are transitions zones, examples of this are present in [Figure 4.6](#) [Figure 4.5](#). The tiles in transition zone contain palms but are not the most prominent object therefore the model doesn't classified them as palms. Other items that the model tends to wrongly classify are tiles with small bush like plants as they have a similar structure to palms in developing stages, tiles of palms mixed with forest and other palms in different developing stages, tiles with bushes or plants that have a similar structure to palms. This is notable in transition zones where isolated palms can be confused with other types of plants like bushes. The performance of the model lead us to the conclusion that features extracted from the images that contain contrasting objects (palm, dirt roads) where not conclusive enough for the machine learning models to properly classify.

This method was designed with the idea of being simple, so that it would be easy to apply in other context and by different users, prove of this is, that all the workflow was performed on a personal computer with the following characteristics: i7-6700HQ CPU @ 2.60GHz and 8 GB of DDR 3 RAM

The main limitation of this method is the small data set, gathering more information is recommended and necessary to create a more robust method that will be able to generalize and adapt to other tasks. A bigger data set with labels defining objects as palms, could be used to train a CNN to detect palms in the images. The image input size for `OerFeat`, defined the size of the tiles that would be classified, in general this didn't allow for a single tile to contain only one palm, therefore single palm tree detection was not possible with this method,

This study can serve as a building block for further refining image classification using CNN and machine learning algorithms, next steps that should be considered are creating a large data set to train a neural network from scratch and to fine tune a pre trained CNN for classifying palm trees, both of these approaches could yield better results and would lead to accomplishing other tasks like detecting palms and automatically counting them. As this is a new and fast developing field improvements to methods and models are being tested constantly, therefore trying other architectures to find if there's one that outperforms others would be an interesting follow up study. As well trying other networks that could perform classification and detection tasks with smaller data sets

5.2 Conclusion

This study provides a method using a pre trained CNN as feature extractor coupled with machine learning algorithms for the classification on very high resolution images obtained with a UAV. This study adds to the growing list of UAV applications in a remote sensing context as well to the fast developing field of convolutional neural network for image classification on remotely sensed data.

Extracting features from very high resolution images was successful, as they could be used to classify if an image had palms, they were different enough for machine learning models to classify the features as tiles having palms or not. The ML models did a good task at classifying the features [section 4.5](#). SVM was the model that achieved the highest classification, LR, GBC, KNN all performed at a similar range, the lowest performance was DT. Based on this finding the SVM model was adopted for image classification. The model had problems classifying images that included several types of land cover as compared to images that had only palms or only forest. The application of this method further proves that transfer learning is a viable option for image classification task with small data sets. This further validates this type of methods for remote sensing applications.

BIBLIOGRAPHIC REFERENCES

- Atzberger, C. (2013). "Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs." In: *Remote sensing* 5.2, pp. 949–981.
- Basse, R. M., O. Charif, and K. Bódis (2016). "Spatial and temporal dimensions of land use change in cross border region of Luxembourg. Development of a hybrid approach integrating GIS, cellular automata and decision learning tree models." In: *Applied Geography* 67, pp. 94–108.
- Bazi, Y. and F. Melgani (2018). "Convolutional SVM networks for object detection in UAV imagery." In: *IEEE Transactions on Geoscience and Remote Sensing* 56.6, pp. 3107–3118.
- Belgiu, M. and L. Drăguț (2016). "Random forest in remote sensing: A review of applications and future directions." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 114, pp. 24–31.
- Berteška, T. and B. Ruzgienė (2013). "Photogrammetric mapping based on UAV imagery." In: *Geodesy and Cartography* 39.4, pp. 158–163.
- Blaschke, T., G. J. Hay, M. Kelly, S. Lang, P. Hofmann, E. Addink, R. Q. Feitosa, F. Van der Meer, H. Van der Werff, F. Van Coillie, et al. (2014). "Geographic object-based image analysis—towards a new paradigm." In: *ISPRS journal of photogrammetry and remote sensing* 87, pp. 180–191.
- Bollard-Breen, B., J. D. Brooks, M. R. Jones, J. Robertson, S. Betschart, O. Kung, S. C. Cary, C. K. Lee, and S. B. Pointing (2015). "Application of an unmanned aerial vehicle in spatial mapping of terrestrial biology and human disturbance in the McMurdo Dry Valleys, East Antarctica." In: *Polar biology* 38.4, pp. 573–578.
- Casagrande, G., A. Sik, and G. Szabó (2018). *Small Flying Drones*. Springer.
- Chen, G., Q. Weng, G. J. Hay, and Y. He (2018). "Geographic object-based image analysis (GEOBIA): emerging trends and future opportunities." In: *GIScience & Remote Sensing* 55.2, pp. 159–182.
- Chen, S. W., S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar (2017). "Counting apples and oranges with deep learning: A data-driven approach." In: *IEEE Robotics and Automation Letters* 2.2, pp. 781–788.

- Colomina, I. and P. Molina (2014). "Unmanned aerial systems for photogrammetry and remote sensing: A review." In: *ISPRS Journal of photogrammetry and remote sensing* 92, pp. 79–97.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database." In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, pp. 248–255.
- Eltner, A., A. Kaiser, C. Castillo, G. Rock, F. Neugirg, and A. Abellán (2016). "Image-based surface reconstruction in geomorphometry—merits, limits and developments." In: *Earth Surface Dynamics* 4.2, pp. 359–389.
- Everaerts, J. et al. (2008). "The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping." In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37.2008, pp. 1187–1192.
- Fitzherbert, E. B., M. J. Struebig, A. Morel, F. Danielsen, C. A. Brühl, P. F. Donald, and B. Phalan (2008). "How will oil palm expansion affect biodiversity?" In: *Trends in ecology & evolution* 23.10, pp. 538–545.
- Foody, G. M. and A. Mathur (2004). "A relative evaluation of multiclass image classification by support vector machines." In: *IEEE Transactions on geoscience and remote sensing* 42.6, pp. 1335–1343.
- Fukushima, K. and S. Miyake (1982). "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." In: *Competition and cooperation in neural nets*. Springer, pp. 267–285.
- Gago, J., C. Douthe, R. Coopman, P. Gallego, M. Ribas-Carbo, J. Flexas, J. Escalona, and H. Medrano (2015). "UAVs challenge to assess water stress for sustainable agriculture." In: *Agricultural water management* 153, pp. 9–19.
- Girshick, R. (2015). "Fast R-CNN." In: *The IEEE International Conference on Computer Vision (ICCV)*.
- Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hay, G. J. and G. Castilla (2008). "Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline." In: *Object-based image analysis*. Springer, pp. 75–89.
- Hedley, J. D., C. M. Roelfsema, I. Chollett, A. R. Harborne, S. F. Heron, S. Weeks, W. J. Skirving, A. E. Strong, C. M. Eakin, T. R. Christensen, et al. (2016). "Remote sensing of coral reefs for monitoring and management: a review." In: *Remote Sensing* 8.2, p. 118.
- Hu, F., G.-S. Xia, J. Hu, and L. Zhang (2015). "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery." In: *Remote Sensing* 7.11, pp. 14680–14707.

- Hubel, D. H. and T. N. Wiesel (1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." In: *The Journal of physiology* 160.1, pp. 106–154.
- Hung, C., Z. Xu, and S. Sukkarieh (2014). "Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a UAV." In: *Remote Sensing* 6.12, pp. 12037–12054.
- Iizuka, K., M. Itoh, S. Shiodera, T. Matsubara, M. Dohar, and K. Watanabe (2018). "Advantages of unmanned aerial vehicle (UAV) photogrammetry for landscape analysis compared with satellite data: A case study of postmining sites in Indonesia." In: *Cogent Geoscience* 4.1, pp. 1–15.
- International Civil Aviation Organization (2011). *Unmanned Aircraft Systems (UAS)*. 1, pp. 1–54.
- Karpathy, A. (2016). *Lecture notes Convolutional Neural Networks for Visual Recognition*.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems* 25. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105. URL: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- Lary, D. J., A. H. Alavi, A. H. Gandomi, and A. L. Walker (2016). "Machine learning in geosciences and remote sensing." In: *Geoscience Frontiers* 7.1, pp. 3–10.
- Lary, D. J. (2010). "Artificial intelligence in geoscience and remote sensing." In: *Geoscience and Remote Sensing New Achievements*. InTech.
- Lawrence, R. L. and C. J. Moran (2015). "The AmericaView classification methods accuracy comparison project: A rigorous approach for model selection." In:
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
- Li, W., H. Fu, L. Yu, and A. Cracknell (2017). "Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images." In: *Remote Sensing* 9.1. ISSN: 2072-4292. DOI: [10.3390/rs9010022](https://doi.org/10.3390/rs9010022). URL: <http://www.mdpi.com/2072-4292/9/1/22>.
- Lillesand, T., R. W. Kiefer, and J. Chipman (2014). *Remote sensing and image interpretation*. John Wiley & Sons.
- Long, J., E. Shelhamer, and T. Darrell (2015). "Fully convolutional networks for semantic segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.
- Luna, I. and A. Lobo (2016). "Mapping crop planting quality in sugarcane from UAV imagery: a pilot study in Nicaragua." In: *Remote Sensing* 8.6, p. 500.
- Marmanis, D., M. Datcu, T. Esch, and U. Stilla (2016). "Deep learning earth observation classification using ImageNet pretrained networks." In: *IEEE Geoscience and Remote Sensing Letters* 13.1, pp. 105–109.

- Masini, N., F. Gizzi, M. Biscione, V. Fundone, M. Sedile, M. Sileo, A. Pecci, B. Lacovara, and R. Lasaponara (2018). "Medieval Archaeology Under the Canopy with LiDAR. The (Re) Discovery of a Medieval Fortified Settlement in Southern Italy." In: *Remote Sensing* 10.10, p. 1598.
- Maxwell, A. E., T. A. Warner, and F. Fang (2018). "Implementation of machine-learning classification in remote sensing: An applied review." In: *International journal of remote sensing* 39.9, pp. 2784–2817.
- Megahed, Y., P. Cabral, J. Silva, and M. Caetano (2015). "Land cover mapping analysis and urban growth modelling using remote sensing techniques in greater Cairo region—Egypt." In: *ISPRS International Journal of Geo-Information* 4.3, pp. 1750–1769.
- Mesas-Carrascosa, F. J., I. C. Rumbao, J. A. B. Berrocal, and A. G.-F. Porras (2014). "Positional quality assessment of orthophotos obtained from sensors onboard multi-rotor UAV platforms." In: *Sensors* 14.12, pp. 22394–22407.
- Mikita, T., P. Janata, and P. Surovỳ (2016). "Forest stand inventory based on combined aerial and terrestrial close-range photogrammetry." In: *Forests* 7.8, p. 165.
- Mountrakis, G., J. Im, and C. Ogole (2011). "Support vector machines in remote sensing: A review." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.3, pp. 247–259.
- Navalgund, R. R., V. Jayaraman, and P. Roy (2007). "Remote sensing applications: An overview." In: *Current Science (00113891)* 93.12.
- Nogueira, K., O. A. Penatti, and J. A. dos Santos (2017). "Towards better exploiting convolutional neural networks for remote sensing scene classification." In: *Pattern Recognition* 61, pp. 539–556.
- Othman, E., Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani (2016). "Using convolutional features and a sparse autoencoder for land-use scene classification." In: *International Journal of Remote Sensing* 37.10, pp. 2149–2167.
- Pajares, G. (2015). "Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs)." In: *Photogrammetric Engineering & Remote Sensing* 81.4, pp. 281–330.
- Pal, M. and P. M. Mather (2003). "An assessment of the effectiveness of decision tree methods for land cover classification." In: *Remote sensing of environment* 86.4, pp. 554–565.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Penatti, O. A. B., K. Nogueira, and J. A. dos Santos (2015). "Do Deep Features Generalize From Everyday Objects to Remote Sensing and Aerial Scenes Domains?" In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

- Petropoulos, G. P., C. Kalaitzidis, and K. P. Vadrevu (2012). "Support vector machines and object-based classification for obtaining land-use/cover cartography from Hyperion hyperspectral imagery." In: *Computers & Geosciences* 41, pp. 99–107.
- Pflanz, M., H. Nordmeyer, and M. Schirrmann (2018). "Weed Mapping with UAS Imagery and a Bag of Visual Words Based Image Classifier." In: *Remote Sensing* 10.10, p. 1530.
- Ren, S., K. He, R. Girshick, and J. Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In: *Advances in Neural Information Processing Systems* 28. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 91–99. URL: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>.
- Richter, G. M., F. Agostini, A. Barker, D. Costomiris, and A. Qi (2016). "Assessing on-farm productivity of Miscanthus crops by combining soil mapping, yield modelling and remote sensing." In: *Biomass and Bioenergy* 85, pp. 252–261.
- Rogan, J., J. Franklin, D. Stow, J. Miller, C. Woodcock, and D. Roberts (2008). "Mapping land-cover modifications over large areas: A comparison of machine learning algorithms." In: *Remote Sensing of Environment* 112.5, pp. 2272–2283.
- Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge." In: *International Journal of Computer Vision* 115.3, pp. 211–252.
- Salamí, E., C. Barrado, and E. Pastor (2014). "UAV flight experiments applied to the remote sensing of vegetated areas." In: *Remote Sensing* 6.11, pp. 11051–11081.
- Salberg, A.-B. (2015). "Detection of seals in remote sensing images using features extracted from deep convolutional neural networks." In: *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, pp. 1893–1896.
- Sermanet, P., D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun (2013). "Overfeat: Integrated recognition, localization and detection using convolutional networks." In: *arXiv preprint arXiv:1312.6229*.
- Sharif Razavian, A., H. Azizpour, J. Sullivan, and S. Carlsson (2014). "CNN features off-the-shelf: an astounding baseline for recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813.
- Silva, J. D.B. F. da (2016). "Contributions for the improvement of specific class mapping." Doctoral dissertation. Universidade Nova de Lisboa Information Management School.
- Simonyan, K. and A. Zisserman (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: [1409.1556](https://arxiv.org/abs/1409.1556) [cs.CV].
- Sumathi, S, S. Chai, and A. Mohamed (2008). "Utilization of oil palm as a source of renewable energy in Malaysia." In: *Renewable and Sustainable Energy Reviews* 12.9, pp. 2404–2421.

- Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). "Going Deeper With Convolutions." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tianyang, D., Z. Jian, G. Sibin, S. Ying, and F. Jing (2018). "Single-Tree Detection in High-Resolution Remote-Sensing Images Based on a Cascade Neural Network." In: *ISPRS International Journal of Geo-Information* 7.9, p. 367.
- Villa, T. F., F. Gonzalez, B. Miljevic, Z. D. Ristovski, and L. Morawska (2016). "An overview of small unmanned aerial vehicles for air quality measurements: Present applications and future perspectives." In: *Sensors* 16.7, p. 1072.
- Wang, R., S. Zhang, L. Pu, J. Yang, C. Yang, J. Chen, C. Guan, Q. Wang, D. Chen, B. Fu, et al. (2016). "Gully erosion mapping and monitoring at multiple scales based on multi-source remote sensing data of the Sancha River Catchment, Northeast China." In: *ISPRS International Journal of Geo-Information* 5.11, p. 200.
- Waske, B., J. A. Benediktsson, and J. R. Sveinsson (2009). "Classifying remote sensing data with support vector machines and imbalanced training data." In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 375–384.
- Watts, A. C., V. G. Ambrosia, and E. A. Hinkley (2012). "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use." In: *Remote Sensing* 4.6, pp. 1671–1692.
- Yuan, Y. and X. Hu (2016). "Random forest and objected-based classification for forest pest extraction from UAV aerial imagery." In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 41, p. 1093.
- Zecha, C., J. Link, and W. Claupein (2013). "Mobile sensor platforms: Categorisation and research applications in precision farming." In: *Journal of Sensors and Sensor Systems* 2.1, pp. 51–72.
- Zeiler, M. D. and R. Fergus (2013). *Stochastic Pooling for Regularization of Deep Convolutional Neural Networks*. arXiv: [1301.3557 \[cs.LG\]](#).
- Zeiler, M. D. and R. Fergus (2014). "Visualizing and understanding convolutional networks." In: *European conference on computer vision*. Springer, pp. 818–833.
- Zhang, L., L. Zhang, and B. Du (2016). "Deep learning for remote sensing data: A technical tutorial on the state of the art." In: *IEEE Geoscience and Remote Sensing Magazine* 4.2, pp. 22–40.
- Zhang, L., Z. Shi, and J. Wu (2015). "A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8.10, pp. 4895–4909.
- Zhou, P., G. Cheng, Z. Liu, S. Bu, and X. Hu (2016). "Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping." In: *Multidimensional Systems and Signal Processing* 27.4, pp. 925–944.

- Zhu, X. X., D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer (2017). "Deep learning in remote sensing: a comprehensive review and list of resources." In: *IEEE Geoscience and Remote Sensing Magazine* 5.4, pp. 8–36.
- Zortea, M., M. Nery, B. Ruga, L. B. Carvalho, and A. C. Bastos (2018). "Oil-Palm Tree Detection in Aerial Images Combining Deep Learning Classifiers." In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 657–660.



ANNEX

This section contains information and results that were not presented in the main text but are complementary to the results

I.1 Example of features extracted using OverFeat

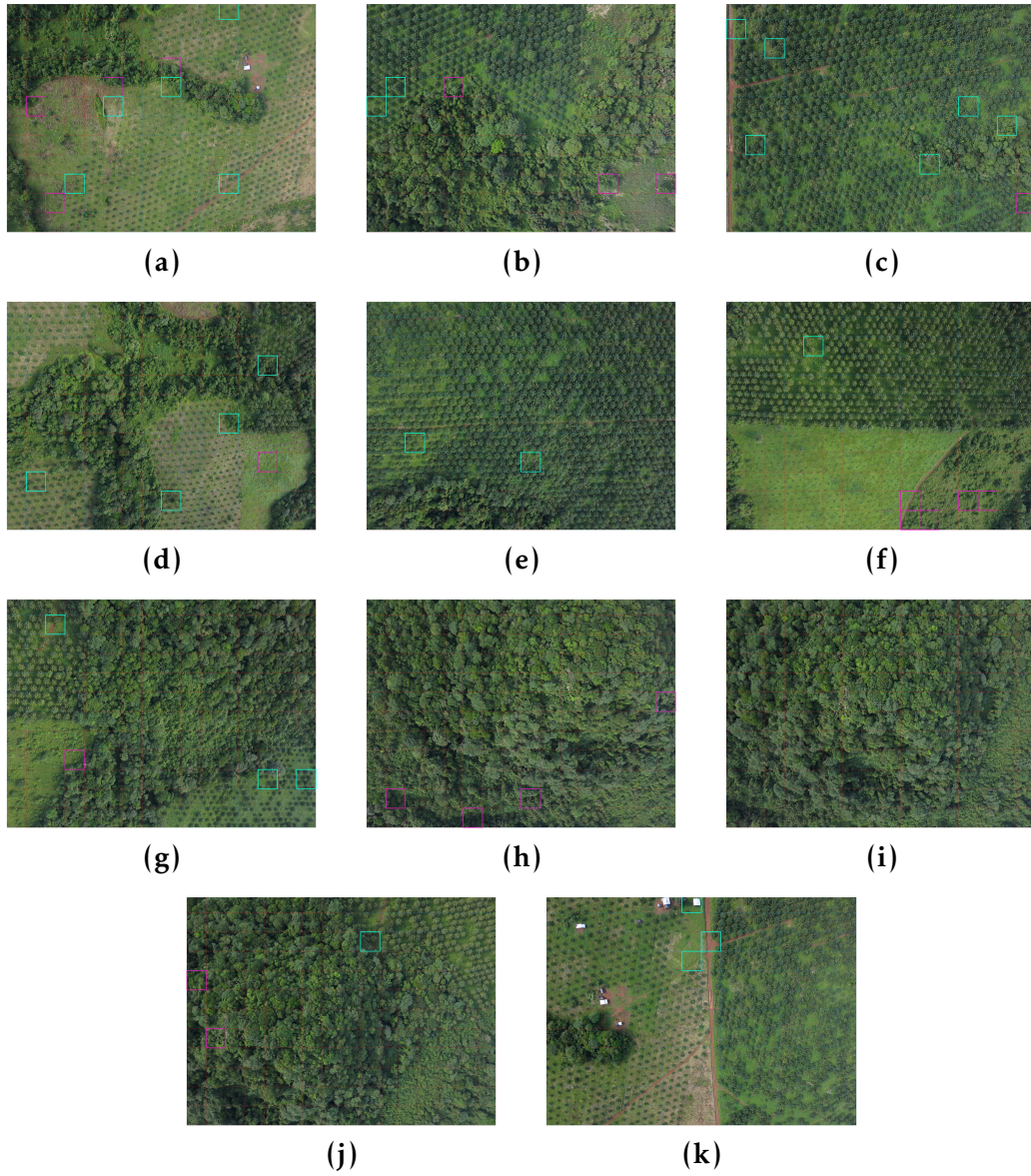
Only a portion of the features are presented since each feature has 4096 variables. The first number indicates the label, followed by the values of the features.

- Example of feature extracted with OverFeat for a tile label as "1"
1 -0.977721 -0.254899 -0.350732 -2.20161 0.167981 0.032499 -1.24036 -0.368897
0.0988937 -0.26209 -0.931044 0.121792 -0.182609 -1.45426 -1.85707 -0.254737
-1.17318 -0.861829 -2.39979 -2.0163 -0.294961
- Example of feature extracted with OverFeat for a tile label as "0"
0 -2.67815 -1.83985 -0.732881 -3.21898 -1.22884 -1.39543 -2.83617 -1.52186
-2.04997 -0.404118 -1.40013 -1.29628 -0.978714 -2.79572 -0.372814 -1.82225
-1.96431 -1.44416 -3.20762 -2.65322 -0.602951 -2.64013

I.2 Image classification results

Image classification results from SVM model, all the image are presented.

Figure I.1: Image classification results

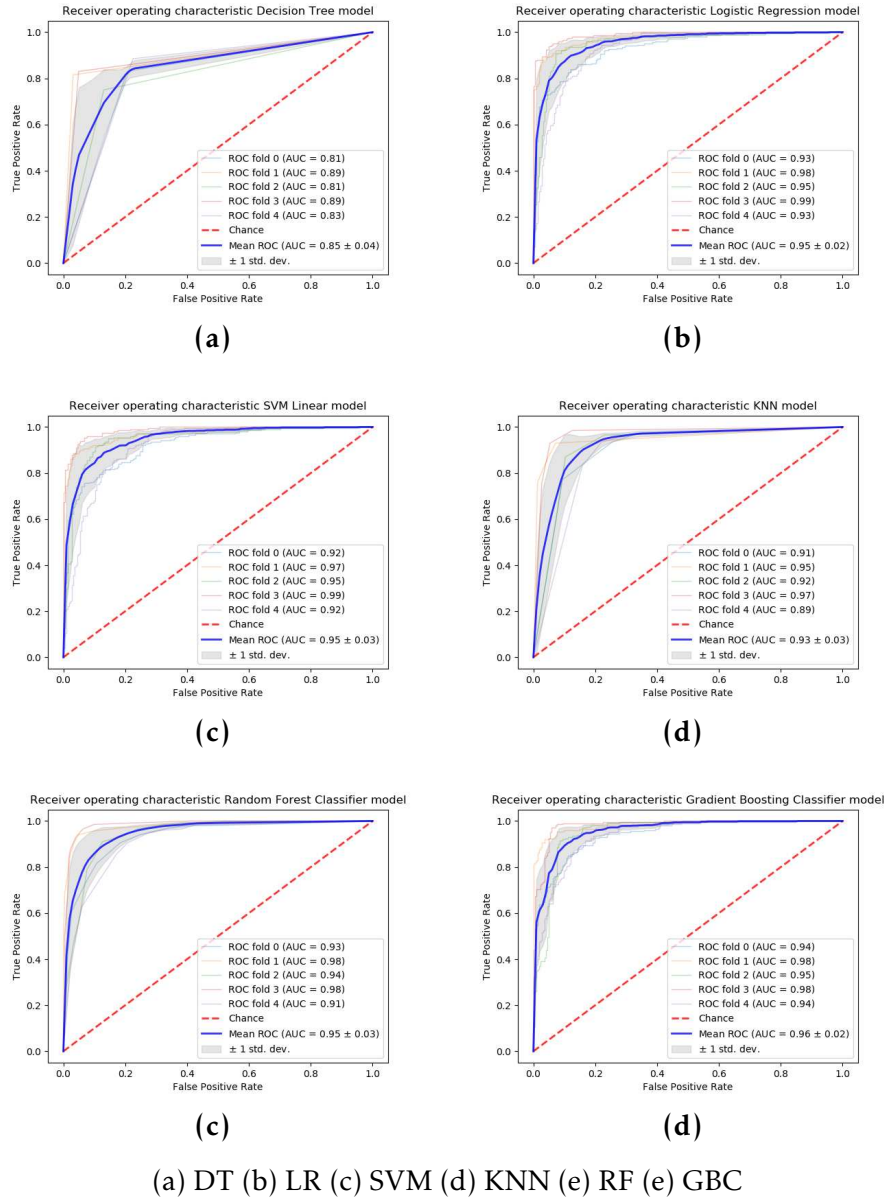


(a) Image 2389 (b) Image 2402 (c) Image 2395 (d) Image 2367 (e) Image 2361 (f) Image 2371 (g) Image 2374 (h) Image 2376 (i) Image 2377 (j) Image 2378 (k) Image 2392

I.3 ROC graphs

ROC and area under the curve graphs for each model.

Figure I.2: ROC results for ML models



I.4 Confusion matrix

Confusion matrices for the classifiers used in the study

Table I.1: Confusion matrix SVM

	SVM	Reference		Total Result
		0	1	
Predicted	0	978	144	1117
	1	133	857	995
	Total Result	1111	1001	2112

Table I.2: Confusion matrix DT

	DT	Reference		Total Result
		0	1	
Predicted	0	978	144	1122
	1	133	857	990
	Total Result	1111	1001	2112

Table I.3: Confusion matrix LR

	LR	Reference		Total Result
		0	1	
Predicted	0	1061	37	1098
	1	50	964	1014
	Total Result	1111	1001	2112

Table I.4: Confusion matrix GBC

	GBC	Reference		Total Result
		0	1	
Predicted	0	1023	67	1090
	1	88	934	1022
	Total Result	1111	1001	2112

Table I.5: Confusion matrix KNN

	KNN	Reference		Total Result
		0	1	
Predicted	0	1030	42	1072
	1	81	959	1040
	Total Result	1111	1001	2112

Table I.6: Confusion matrix RF

	RF	Reference		Total Result
		0	1	
Predicted	0	963	53	1016
	1	148	948	1096
	Total Result	1111	1001	2112



ANNEX 2

Examples of scripts used in the processing of the data are presented here. Image cropping, feature extraction and the machine learning models are all completely presented in this section. A example of the fine tuning and ROC analysis are presented for the SVM model, the rest of the scripts are in these link: <https://github.com/coffeebread42/palm-tree-classification.git>

II.1 Cropping Images

```
1  #!/bin/bash
2
3  img_dir=../data/original
4  out_dir=../data/processed
5  w=250
6
7  echo Start
8
9  n=$(ls -l "$img_dir" | wc -l)
10 n="$((n-1))"
11
12 echo Directory "$img_dir" has "$n" files
13
14 i=1
15 for f in "$img_dir"/*
16 do
17     img=$(echo ${f##*/} | cut -d"." -f1)
18     [ -d "$out_dir"/"$img" ] && rm -r "$out_dir"/"$img"
19     mkdir "$out_dir"/"$img"
20     convert "$f" -crop "$w"x"$w" +repage "$out_dir"/"$img"/"$img"_"$w"x"$w"_%03d.jpg
21     echo "Image $i of $n done"
22     i="$((i+1))"
23 done
24
25 echo Done
```

Listing II.1: Cropping images

II.2 Extracting features with Overfeat

```
1  #!/bin/bash
2
3  extractor=../overfeat/bin/linux_64/overfeat
4  img_dir=../data/processed
5  feat_dir=../data/features
6  out_file=features.data
7
8  #echo Start
9
10 output_file="$feat_dir"/"$out_file"
11
12 touch "$output_file"
13
14 i=0
15 for d in "$img_dir"/*
16 do
```

```
17 #echo Processing image "$d"
18 for f in "$d"/*
19 do
20     #echo Extracting features of "$f"
21     "$extractor" -f "$f" | sed -n 2p >> $output_file
22     #echo Features of "$f" extracted
23     ff=$(echo "$f" | cut -d "/" -f5)
24     echo "$i $ff"
25     ((i++))
26 done
27 #echo Image "$d" is done
28 done
29
30 #echo Done
```

Listing II.2: Feature extraction

II.3 Machine learning Models

```

1  # Load libraries
2  from pandas import read_csv
3  from pandas.tools.plotting import scatter_matrix
4  import numpy as np
5  from matplotlib import pyplot
6  from sklearn.model_selection import train_test_split
7  from sklearn.model_selection import KFold
8  from sklearn.model_selection import cross_val_score
9  from sklearn.metrics import classification_report
10 from sklearn.metrics import confusion_matrix
11 from sklearn.metrics import accuracy_score
12 from sklearn.linear_model import LogisticRegression
13 from sklearn.tree import DecisionTreeClassifier
14 from sklearn.neighbors import KNeighborsClassifier
15 from sklearn.ensemble import RandomForestClassifier
16 from sklearn.ensemble import GradientBoostingClassifier
17 from sklearn.svm import SVC
18 #data directory
19 inf = "/home/roberto/Desktop/work/src/classifiers/flv2.out"
20 ### Input
21 filename = inf
22 ### read data as np array
23 data = np.genfromtxt(filename, delimiter=' ')
24 #y is label column x is the variable columns
25 y = data[:, 0]
26 X = data[:, 1:]
27 #split the datasets
28 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = None)
29 # Spot-Check Algorithms
30 models = []
31 models.append(('LR', LogisticRegression(C= 0.010)))
32 models.append(('RFC', RandomForestClassifier(max_depth= 7,
33                                             n_estimators= 500,
34                                             min_samples_leaf= 0.2,
35                                             min_samples_split= 0.58)))
36 models.append(('KNN', KNeighborsClassifier(n_neighbors=3, p=3)))
37 models.append(('DTC', DecisionTreeClassifier(max_depth=5, min_samples_split=0.3,
38                                             ↪ min_samples_leaf=0.3)))
39 models.append(('GBC', GradientBoostingClassifier(learning_rate= 0.1,
40                                             n_estimators=40,
41                                             max_depth=3,
42                                             min_samples_split=0.6,
43                                             min_samples_leaf=0.35)))
44 models.append(('SVM', SVC(C=10000, kernel='linear')))
45 # evaluate each model in turn
46 results = []
47 names = []
48 for name, model in models:

```

```

48     kfold = KFold(n_splits=10, random_state= None)
49     cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='accuracy')
50     results.append(cv_results)
51     names.append(name)
52     msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
53     print(msg)
54 # Compare Algorithms
55 fig = pyplot.figure()
56 fig.suptitle( 'Classifier Comparison')
57 ax = fig.add_subplot(111)
58 pyplot.boxplot(results, showfliers=False)
59 ax.set_xticklabels(names)
60 pyplot.savefig("classifier_comp3.png")
61 pyplot.show()

```

Listing II.3: Machine Learning Models

II.4 Fine tuning Machine learning Models

II.4.1 Support Vector Machine

```

1  import numpy as np
2  from scipy import interp
3  import matplotlib.pyplot as plt
4  from sklearn.svm import SVC
5  from sklearn.metrics import roc_curve, auc
6  from sklearn.model_selection import StratifiedKFold
7
8  #####
9  # Data IO and generation
10 filename = "flv2.out"
11
12 data = np.genfromtxt(filename, delimiter=' ')
13 y = data[:, 0]
14 X = data[:, 1:]
15 n_samples, n_features = X.shape
16 random_state = np.random.RandomState(0)
17 # Run classifier with cross-validation and plot ROC curves
18 cv = StratifiedKFold(n_splits=5)
19 classifier = SVC(kernel='linear', C = cost, probability=True,
20                 random_state=random_state)
21
22 tprs = []
23 aucs = []
24 mean_fpr = np.linspace(0, 1, 100)
25
26 i = 0
27 for train, test in cv.split(X, y):
28     probas_ = classifier.fit(X[train], y[train]).predict_proba(X[test])

```

```

29     # Compute ROC curve and area the curve
30     fpr, tpr, thresholds = roc_curve(y[test], probas[:, 1])
31     tprs.append(interp(mean_fpr, fpr, tpr))
32     tprs[-1][0] = 0.0
33     roc_auc = auc(fpr, tpr)
34     aucs.append(roc_auc)
35     plt.plot(fpr, tpr, lw=1, alpha=0.3,
36              label='ROC fold %d (AUC = %0.2f)' % (i, roc_auc))
37
38     i += 1
39 plt.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',
40          label='Chance', alpha=.8)
41
42 mean_tpr = np.mean(tprs, axis=0)
43 mean_tpr[-1] = 1.0
44 mean_auc = auc(mean_fpr, mean_tpr)
45 std_auc = np.std(aucs)
46 plt.plot(mean_fpr, mean_tpr, color='b',
47          label=r'Mean ROC (AUC = %0.2f $\pm$ %0.2f)' % (mean_auc, std_auc),
48          lw=2, alpha=.8)
49
50 std_tpr = np.std(tprs, axis=0)
51 tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
52 tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
53 plt.fill_between(mean_fpr, tprs_lower, tprs_upper, color='grey', alpha=.2,
54                  label=r'$\pm$ 1 std. dev.')
55
56 plt.xlim([-0.05, 1.05])
57 plt.ylim([-0.05, 1.05])
58 plt.xlabel('False Positive Rate')
59 plt.ylabel('True Positive Rate')
60 plt.title('Receiver operating characteristic SVM Linear model')
61 plt.legend(loc="lower right")
62 plt.savefig("svm_linear_roc.png")
63 plt.show()

```

Listing II.4: SVM ROC

```

1  import numpy as np
2  from sklearn import svm
3  from sklearn.metrics import roc_curve, auc
4  import matplotlib.pyplot as plt
5  from matplotlib.legend_handler import HandlerLine2D
6
7  inf = "/home/roberto/Desktop/work/src/classifiers/flv2.out"
8  ### Input
9  filename = inf
10 ### Process
11 data = np.genfromtxt(filename, delimiter=' ')
12

```

```
13 y = data[:, 0]
14 X = data[:, 1:]
15
16 kernels = ['linear', 'rbf', 'poly']
17 for kernel in kernels:
18     svc = svm.SVC(kernel=kernel).fit(X,y)
19     plotSVC('kernel=' + str(kernel))
20
21 def plotSVC(title):
22     #create a mesh to plot in'
23     x_min, x_max = X[:,0].min() - 1, X[:,0].max() + 1
24     y_min, y_max = X[:,1].min() - 1, X[:,1].max() + 1
25     h = (x_max / x_min)/100
26     xx,yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
27     plt.subplot(1, 1, 1)
28     Z = svc.predict(np.c_[xx.ravel(), yy.ravel()])
29     Z = Z.reshape(xx.shape)
30     plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.8)
31     plt.scatter(X[:, 1:], c=y, cmap=plt.cm.Paired)
32     plt.xlabel('vector')
33     plt.ylabel('label')
34     plt.xlim(xx.min(), xx.max())
35     plt.title(title)
36     plt.show()
```

Listing II.5: SVM Fine Tuning



Masters Program in **Geospatial Technologies**



Supported by:



Education and Culture
ERASMUS MUNDUS

